

*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ*

КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ: ОСВІТА, НАУКА, ВИРОБНИЦТВО

НАУКОВИЙ
ЖУРНАЛ



Відповідальний редактор – проф., д-р техн. наук В. Д. Рудь

№30-31 2018

*м. Луцьк
Видавництво Луцького національного технічного університету*

КОМП'ЮТЕРНО-ІНТЕГРОВАНІ
ТЕХНОЛОГІЇ:
ОСВІТА, НАУКА, ВИРОБНИЦТВО

№30-31 2018р.

РЕДАКЦІЙНА КОЛЕГІЯ:

Головний редактор:

проф., д.т.н. Рудь В.Д. (м.Луцьк)

Заступники головного редактора:

доц., к.т.н. Мельник К.В. (м.Луцьк)

доц., к.т.н. Герасимчук О.О. (м.Луцьк)

Відповідальний секретар:

мол.наук.співробітник Свиридюк К.А. (м.Луцьк)

Члени редакційної колегії:

проф, PhD. Milosz Marek (Польща, м.Люблін)

проф, PhD. Alison McMillan (Великобританія)

проф, PhD. Дехтяр Ю.Д. (Литва, м.Рига)

проф., д.т.н. Мазур М.П. (м.Хмельницьк)

проф., д.т.н. Мельник А.О. (м.Київ)

проф., д.т.н. Сидорчук О.В. (м.Київ)

проф., д.т.н. Тарасенко В.П. (м.Київ)

проф., д.ф-м.н. Пастернак Я.М. (м.Луцьк)

проф., д.т.н. Андрушак І.Є. (м.Луцьк)

проф., д.т.н. Делявський М.В. (м.Луцьк)

проф., д.е.н. Рудь Н.Т. (м.Луцьк)

проф., д.т.н. Пальчевський Б.О. (м.Луцьк)

доц., к.т.н. Драган О.В. (м.Брест, Білорусія)

доц., к.т.н. Лотиш В.В. (м.Луцьк)

доц., к.т.н. Гуменюк Л.О. (м.Луцьк)

доц., к.т.н. Пех П.А. (м.Луцьк)

доц., к.т.н. Самчук Л.М. (м.Луцьк)

доц., к.пед.н. Потапюк Л.М. (м.Луцьк)

доц., к.т.н. Решетило О.М. (м.Луцьк)

доц., к.т.н. Повстяной О.Ю. (м.Луцьк)

Адреса редколегії:

Луцький національний технічний університет,
кафедра комп'ютерної інженерії.

вул.Львівська 75, ауд.141

м.Луцьк, 43018

тел. (0332) 74-61-15

E-mail: cit@lntu.edu.ua,

сайт журналу: **ki.lutsk-ntu.com.ua**

Журнал засновано у грудні 2010 р.
Свідоцтво про реєстрацію КВ № 16705–5277 Р.
Засновник: Луцький національний технічний університет

Рекомендовано до друку Вченою радою
Луцького національного технічного університету
(протокол №8 засідання від 27.03.2018)

Журнал рішенням МОН України
наказом №515 від 16.05.2016р,
включено в перелік наукових фахових видань.
Журнал має російський індекс наукового цитування
(РІНЦ)

Індексується в міжнародних базах даних:

Universal Impact Factor,
Open Academic Journals Index.

ISSN 2524-0560 (Online)

ISSN 2524-0552 (Print)

ЗМІСТ

ІНФОРМАТИКА ТА ОБЧИСЛЮВАЛЬНА ТЕХНІКА

Багнюк Н.В., Михайлуца П., Христинець А.О. Web авторизація для користувачів локальної мережі. <i>[Nataliia Bahniuk, Pavlo Mykhailutsa, Andrii Khrystinets .Web service authorization for local network users.]</i>	5
Бортник К.Я., Ломінська Г.Ю. Технології аналізу наслідків кібератак.	10
Бортник К.Я., Ольшевський О.В., Пащук В.Ю. Інтернет речей та як він змінить наше життя у майбутньому.	14
Гненний А.П., Гордієнко Ю.Г. Актуальні механізми ідентифікації особистості в інтернеті речей.	19
Гринюк С.В., Поліщук М.М., Міскевич О.І., Харковець Р.В. Програмне забезпечення для шифрування та дешифрування інформації криптографічними методами засобами Visual Studio.	26
Губаль Г.М. Використання деяких команд мови L ^A T _E X для створення математичних текстів.	32
Журавська І.М. IoT-мережа на базі Bluetooth-модулів для автоматизованого керування споживанням енергоресурсів. <i>[Zhuravska I.M. IoT network based on Bluetooth modules for automatized control power consumption.]</i>	37
Заяць В.М., Маєвський Я., Марціняк Т., Рибицька О.М., Заяць М.М. Моделі та алгоритми для опрацювання нечіткої інформації. <i>[Zaiats V.M., Majewski J., Marciniak T., Rybytska O.M., Zaiats M.M. Models and algorithms for processing of fuzzy information.]</i>	45
Здолбіцька Н.В., Мельник Д.С. Створення мобільного додатку в IDE Android Studio для діагностики смартфонів на основі операційної системи Android.	52
Коломоєць Г.П. Програма підготовки до друку дипломів про вищу освіту за даними ЄДЕБО із використанням сучасних технологій Java.	57
Костючко С.М., Дацюк С.В., Нагорнюк А.І. Сесії та авторизація користувачів інтернет-магазину за допомогою PYTHON та DJANGO.	66
Лавренчук С.В., Ілюшик Р.С. Переваги створення real-time додатків на базі асинхронної бібліотеки Channels та фреймворка Django.	70
Лисенко С.М., Гайбура Ю.О., Стецюк В.М. Метод виявлення троянських програм на основі апарату нечіткої кластеризації.	75
Мельник В.М., Багнюк Н.В., Нагорнюк А.І. Особливості робота з базами даних у фреймворку Django.	83
Мельник В.М., Мельник К.В., Шульга Б.В. Порівняння трьох найпопулярніших web-каркасів для розробки проектів в Python.	88
Мельник К.В., Мельник В.М., Мацібора А.С. Дослідження контентних методів розпізнавання спаму	95
Міскевич О.І., Сичов Д.І., Христинець А.О. Про модернізацію локально-обчислювальної мережі на ПрАТ “Волиньобленерго” на основі GRE-tunnel з використанням шифрування IPSec.	100
Муляр В. П. Основи розробки додатків з використанням технології JavaFX.	104
Пех П.А., Кобук Р.Ф. C++Builder програмно-демонстраційний комплекс для дослідження рівнянь з параметрами методом перетину прямими лініями.	111
Пех П.А., Тихомиров В.В. Методи формування системних вимог до АЗ для громіздких 3D програм.	120
Попов Д.А., Костенко В.Л., Ядрова М.В., Кондратьєв С.Б. Разработка мультисенсорного сонара для слепых на основе микроконтроллера STM8L.	126
Савенко О.С., Грибинчук В.І., Кульчицький М.О. Архітектура багаторівневої програмної системи виявлення шкідливого програмного забезпечення в локальних комп’ютерних мережах.	132
Струзік В.А., Грибков С. В., Литвин А. О. Дослідження методів і підходів проведення рефакторингу баз даних.	141

Христинець Н.А., Багнюк Н.В., Матейчук Н.В., Дзюбинський В.В. Аналіз PHP-шаблонізаторів та використання Twig у Drupal 8.	146
Чорнобай К. Ю., Грибков С.В. Розробка бюджетної системи біометричної ідентифікації з використання нейронних мереж.	151
Яцюк С. М., Яцюк А. В., Муляр В. П. Використання фреймворків для розробки Web-додатків на мові PHP.	156

АВТОМАТИКА ТА УПРАВЛІННЯ

Андрушак І.Є., Матвійв Ю.Я., Кошелюк В.А., Сіваковська О.М., Марценюк В.П. Визначення взаємовпливу одночасно утворених тріщин на модуль вектора переміщень. [I.Ye.Andrushchak, Yu.Ya. Matthew, V.A. Koshelyuk, O.M. Sivakovskaya, V.P. Martsenyuk. Definition of interaction at the same time attended tracks on the module of the vector of the transition.]	160
Бомба А.Я., Федонюк А. А. Про узагальнення однієї моделі інформаційної боротьби.	165
Гінайло П.І., Лісковець С.М., Тимошук В.М., Грінченко Л.Г. Необхідні умови екстремуму для лінійних багатозначних відображень.	171
Журило А. Г., Сівак Є. М. Деякі питання щодо невідповідності стандартів ЄСКД та ISO.	176
Захарова Л.М. Розробка алгоритму ідентифікації дисипативних структур й визначення їх паттернів.	181
Ісмаїлова Н.П., Трушков Г.В. Геометричне моделювання просторового параметричного кінематичного гвинта	187
Кириченко А.М., Аль Ібрахімі Метак М.А. Робочий простір та особливості 5-осьового паралельного верстата з надлишковими приводами.	196
Кізим С. О., Ройко О. О., Юхта О. А. Інформаційно-комунікаційні технології в самостійній роботі студентів.	201
Коцюба А.Ю., Сітовський В.О. Реалізація теорії клітинних автоматів в попиксельній обробці растрової графіки.	207
Крайник Я.М., Перов В.О. Організація скінченного автомату управління процесом декодування для декодера Turbo-Product-кодів на базі FPGA.	213
Кузьмич О.І., Маркіна Л.М., Якимчук Н.М. Моделювання та розробка систему керування процесом змішування на базі процесу водно-теплової обробки при виробництві спирту.	220
Лишук В.В., Селепина Й.Р., Ящинський Л.В., Костючко С.М. Застосування методу Ейлера в задачах динаміки.	227
Лісковець С.М., Гуда О.В. Професійне спрямування курсу «Вища математика» для студентів економічного профілю.	232
Матвійв Ю.Я., Андрушак І.Є., Гануліч Б.К., Крадінова Т.А. Практичне застосування критерію для визначення довготривалої міцності та залишкової довговічності елементів конструкцій за низькотемпературної повзучості. [Matviyiv Y.Ya., Andrushchak I.E, Ganulich B.K., Kradinova T.A. Practical application of the criterion for determining the long-term strength and residual durability of structural elements for low-temperature creep.]	238
Молнар О.О., Бан Г.Й., Гал Д.Л., Герасимов В.В. Система телеметрії з елементами доповненої реальності для співробітників служб порятунку на основі ПЛІС.	244
Поліщук М.М., Гринюк С.В., Білоус М.В. Лабораторний блок живлення на базі мікроконтролера STM32F103VET6.	251
Поплавська Г.В. Напрямки впровадження нових інформаційних технологій у процесі контролю та оцінки знань студентів.	257
Ройко Л.Л., Ройко О.О. Прикладна спрямованість курсу “Математика для економістів та економічне моделювання”.	263
Филипчук В.Л., Филипчук Л.В., Шаталов О.С. Безпека експлуатації автоматизованих очисних споруд для знешкодження стічних вод з токсичними окисно-відновними домішками.	269
Черніков М.М., Куваєва В.І., Болтънков В.О. Система розподіленої колективної експертизи з захистом від фальсифікацій.	276
Юхта А.О., Кізим С.О., Ройко О.О., Ройко О.Ю. Аналіз онлайн-алгоритмів на прикладі задачі про прокат лиж з двома опціями.	281

УДК 004.023

PhD Nataliia Bahniuk, Pavlo Mykhailutsa, Andrii Khrystinets.
Lutsk National Technical University

WEB SERVICE AUTHORIZATION FOR LOCAL NETWORK USERS

Багнюк Н.В. к.т.н., Михайлуца П., Христинець А.О. Web авторизація для користувачів локальної мережі. У статті було проаналізовано основні методи аторизації для web. Дані методи використовуються для процесу розробки та побудови логічної архітектури web сайтів. Основною задачею аналізу було знайти шлях безпечної інтеграції користувачів локальної (корпоративної) мережі у web авторизацію, та надати можливість входу з використанням персональних локальних даних на web сайт. Усі методи передбачають лише структуровані розробки, що базуються на серверній та фронтальній частині, у більшості випадків серверна частина використовується для обслуговування API запитів фронтальної частини.

Багнюк Н.В. к.т.н., Мыхайлуца П., Христинец А.О. Web авторизация пользователей локальной сети. В статье было проанализировано основные методы авторизации для web. Эти методы используются для процесса разработки и построения логической структуры web сайтов. Основной задачей анализа было найти способ безопасной интеграции пользователей локальной (корпоративной) сети в web авторизацию, и предоставить возможность входа с персональными локальными данными на web сайте. Все методы предусматривают только структурированные разработки, которые базируются на серверной и фронтальной частях, в большинстве случаев серверная часть используется для обслуживания API запросов от фронтальной части.

PhD Nataliia Bahniuk, Pavlo Mykhailutsa, Khrystinets A.O. Web service authorization for local network users. In article was analyzed main methods of authorization for web service. These methods are using for web site developing process and architecture logical schema building. Main analysis point is to find the security way to integrate local network (corporate) users to web service authorization, and give a chance to login with personal local credentials into web site. All methods covers only structured builds, which based on backend and frontend, in most cases backend is using to serve API requests from frontend part.

Formulation of scientific problems.

Integration of local network users in to global surface creates a many security requirements. One of the main security issue is to lead all implementation to a few most reliable points: integrity, confidentiality, availability. In order to prior is to analyze all sides of login process for web service and create protected module for implementing local users authorization. Only accurate and complete authorization logic for web services can solve the security problems.

Analysis of research.

Questions about security models for web-based applications and formulation of how to build web service as application discovered in[2]. In this article scientists shows how to build models in order to modern security requirements.

Second well known work in security science that can be used for authorization service development is and[3] republished few times later in 2005, 2008 and 2011 [4].

The last but not the least is[5].

One of the newest article is[6] This invent was published in 2018-01-02 and gained a grant from US government. Good core understanding how to deal with network users and groups are showing in [7]. Same as previous invent gain US government grant and Google patent for future invent [8].

Presentation of the main material and the justification of the results.

There is many type of authentication for web service or sites.

HTTP Basic authentication is a method for the client to provide a username and a password when making a request. This is the simplest possible way to enforce access control as it doesn't require cookies, sessions or anything else. To use this, the client has to send the Authorization header along with every request it makes. The username and password are not encrypted (pic. 1.). There is only one way to up security level of these requests – just to use encrypted connection with SSL/TLS, but if a website uses weak encryption, or an attacker can break it, the usernames and passwords will be exposed immediately. There is no way to log out the user using HTTP Basic authentication.



```

Request Headers
:host: www.linkedin.com
:method: GET
:path: /home?trk=nav_responsive_tab_home
:schema: https
:version: HTTP/1.1
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
accept-encoding: gzip, deflate, sdch
accept-language: en-US,en;q=0.8,hu;q=0.6,nl;q=0.4,es;q=0.2,fr;q=0.2,de;q=0.2
cookie: bcookie="v=268b44d748-8d82-46a4-8577-36c00172794b"; bcookie="v=1620151230722494b3c36f8-q94e-4796-8b49-a5371e6804d1EAQEWbrseJHuyZDHTfipryvThp8h7o"; l1de=108b50b0; _gat=1; L
i_s=906114f5; visit=i=v&M"; _ga=GA1.2.96221765.1448263372; oz_props_fset_size1_undefined=undefined; wutan=14YA1rjunkpRvRnKTLFolTCBA9+AOYr016eMu+s84; sl=v=v161-LQe"; li_at=AQEDAR
uYmFnKfE5AAABUM7yS5AAAFR6m0L04AHZ3TRdQVEG7i06z6HWQZKXW0KcUax8VpTfUf8RaQk1YzYoXaz047r3u4tP28mIy5bTs3GLWoyNsXg9XQp0eT3lwdy3P5pLNICgTw0jCso0a; JSESS10NID="ejax:8745093687238714
270"; liaptrue; l1dc=v=TB16;q=272;u=1:1448263557;t=1448349780;s=AQH_ZUYE278NfLub0ZC21f0atKXlxy"; RT=s=14482635594206;s=https3Ak2F2Fwww.linkedin.com%2Fhome%2F%3Ftrk3dhb_sgn
in; share_setting=PUBLIC; _lipt=0_1bmJ3G5dPtMK4q9DLKnuLXWv5-1W3z3LKS5ImXmJDL9otamnhKTL_Tp4xis1TWowMexz07fMN_z7bLH0dRNXvgr9M7Zu7acgFT4sQG3EFD3sF9zVmVhYnVd9j1IatUYh3z8BfPe0LeRw
d8; lang=v=256lang=en+74s; sds=1%3A151shNdxL36wZcAPaWv581uK3D

```

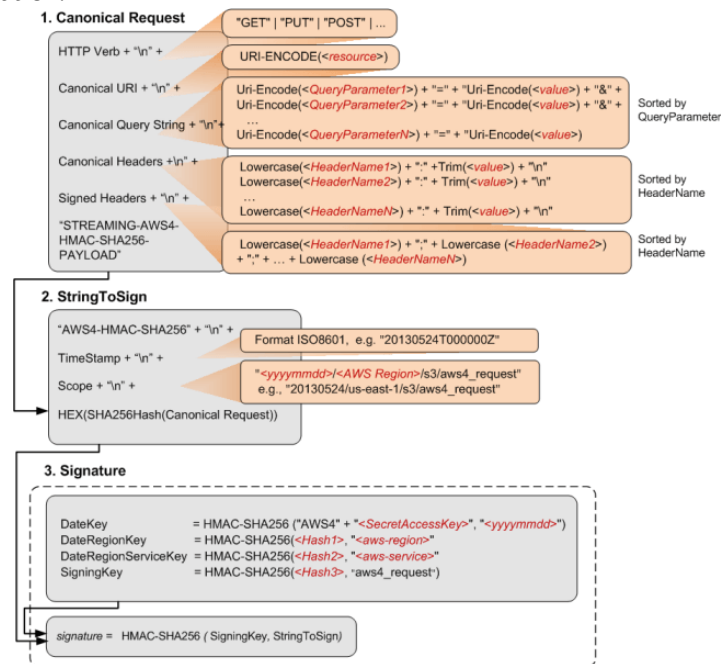
Tokens is most popular authentication logic for now, 90% of newly created web services or sites based on this type of authentication. Tokens in basics is just a generated long string value. For generation we can use different algorithms, such as HASH or MD5 function, or even more crypto-algorithms with different keys length. For example, one of most popular services for generating token is JWT (JSON Web Token) (pic. 3.). There are many different modules (packages) for different languages already created and free.



One-Time passwords algorithms generate a one-time password with a shared secret and either the current time or a counter. These methods are used in applications that leverage two-factor authentication: a user enters the username and password then both the server and the client generates a one-time password. The main problem for this type of authentication with the shared-secret (if stolen) user tokens can be emulated.

© Багнюк Н.В., Михайлуца П., Христинець А.О

this kind of structure is to serve high level of integrity. There is no secret that in different organizations and different companies are using different kind of authorization servers. For example for company with Windows main technology (it means most PCs and server operate by Windows OSs) authentication process will be driven by Active Directory with Kerberos authorization, but for Linux or Unix based networks the leader will be POSIX authorization.



Picture 4. – Signature based authentication

Solving the problem of cross-platform networks is hiding in implementing server structure what can include Kerberos, POSIX and many others authentication logic. On this stage, the leader will be integrating into the local network LDAP (Lightweight Directory Access Protocol) server.

A client starts an LDAP session by connecting to an LDAP server, called a Directory System Agent (DSA), by default on TCP and UDP port 389, or on port 636 for LDAPS (LDAP over SSL, see below). The client then sends an operation request to the server, and the server sends responses in return. With some exceptions, the client does not need to wait for a response before sending the next request, and the server may send the responses in any order. In addition the server may send "Unsolicited Notifications" that are not responses to any request, e.g. before the connection is timed out.

A common alternative method of securing LDAP communication is using an SSL tunnel. The default port for LDAP over SSL is 636. The use of LDAP over SSL was common in LDAP Version 2 (LDAPv2) but it was never standardized in any formal specification. This usage has been deprecated along with LDAPv2, which was officially retired in 2003. Global Catalog is available by default on ports 3268, and 3269 for LDAPS.

Simplest way to build LDAP server is to create docker-compose with container configuration:

version: '2'

services:

openldap:

image: osixia/openldap:1.1.11

container_name: openldap

environment:

LDAP_LOG_LEVEL: "256"

LDAP_ORGANISATION: "Example Inc."

LDAP_DOMAIN: "example.org"

LDAP_BASE_DN: "dc=example,dc=org"

LDAP_ADMIN_PASSWORD: "admin"

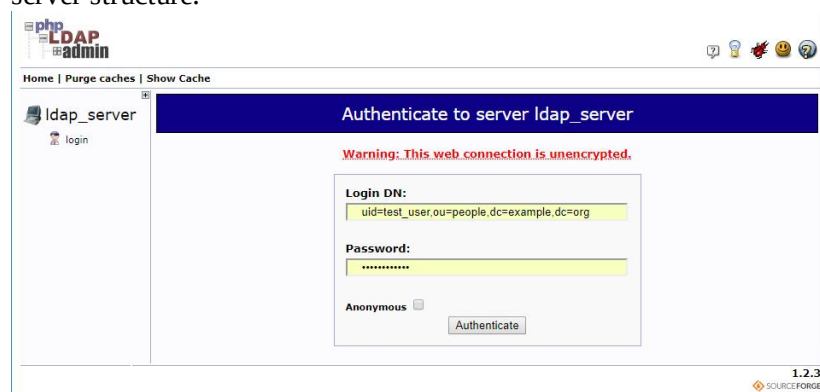
LDAP_CONFIG_PASSWORD: "config"

LDAP_READONLY_USER: "false"

```
#LDAP_READONLY_USER_USERNAME: "readonly"
#LDAP_READONLY_USER_PASSWORD: "readonly"
LDAP_RFC2307BIS_SCHEMA: "false"
LDAP_BACKEND: "hdb"
LDAP_TLS: "true"
LDAP_TLS_CRT_FILENAME: "ldap.crt"
LDAP_TLS_KEY_FILENAME: "ldap.key"
LDAP_TLS_CA_CRT_FILENAME: "ca.crt"
LDAP_TLS_ENFORCE: "false"
LDAP_TLS_CIPHER_SUITE: "SECURE256:-VERS-SSL3.0"
LDAP_TLS_PROTOCOL_MIN: "3.1"
LDAP_TLS_VERIFY_CLIENT: "demand"
LDAP_REPLICATION: "false"
KEEP_EXISTING_CONFIG: "false"
LDAP_REMOVE_CONFIG_AFTER_SETUP: "true"
LDAP_SSL_HELPER_PREFIX: "ldap"
tty: true
stdin_open: true
volumes:
  - /var/lib/ldap
  - /etc/ldap/slapd.d
  - /container/service/slapd/assets/certs/
ports:
  - "389:389"
  - "636:636"
domainname: "example.org"
hostname: "example.org"
```

```
phpldapadmin:
  image: osixia/phpldapadmin:latest
  container_name: phpldapadmin
  environment:
    PHPLDAPADMIN_LDAP_HOSTS: "openldap"
    PHPLDAPADMIN_HTTPS: "false"
  ports:
    - "8080:80"
  depends_on:
    - openldap
```

where openldap service current LDAP container configuration and phpldapadmin web interface (pic. 5) for managing LDAP server structure.



Picture 5. – PHP web interface

All user will looks like ldif file with lines:

```
version: 1
dn: uid=billy,ou=people,dc=example,dc=org
changetype: add
uid: billy
cn: billy
sn: 3
objectClass: top
objectClass: posixAccount
objectClass: inetOrgPerson
loginShell: /bin/bash
homeDirectory: /home/billy
uidNumber: 14583102
gidNumber: 14564100
userPassword: {SSHA}j3lBh1Seqe4rqF1+NuWmjhvtAni1JC5A
mail: billy@example.org
gecos: Billy User
```

In order to docker compose configuration to get access to LDAP server web service should redirect request to one of opened ports 389 or 636. For any manipulation with server web service should to initialize connection to server, than make some operations and close connection. For structured platform with backend and frontend parts any server manipulation are running on backend part, frontend part is using only for getting data and transfer as a request to backend.

All manipulations with LDAP initialization, authorization, manipulation requests represents in python code-base with installed package with basic ldap server functions support (python-ldap==3.0.0):

```
>> import ldap
>> l = ldap.initialize('ldap://localhost')
>> l.protocol_version = '3'
>> l.simple_bind_s('cn=username,dc=example,dc=org', 'password')
```

where 'cn=username,dc=example,dc=org' full user name with it tree position on LDAP server

For automation of any process with python we can use different functions and call them on demand. Successful response of any python-ldap functions will be LDAPObject.

Conclusion and prospects for further research. Any local user with remote access can not be stored on external web-services but we can use python scripts or function in projects (prior backend part) to authenticate local network users. There is no excuses, for low security, and will be a good point, to remember that we can use SSL connection to protect data from stealing. For cryptography issue we can use keys with different length or algorithm. Any new server connection will be protected. Authorization methods presented in this article can provide fast, and protected authentication process for local network users. All represented code was tested on cloud platform and can be used for real web service implementation. LDAP user authorization contains high level of scientific value and can be used for future research.

1. "Mandatory Reporting of Conventional Generation Performance Data" (PDF). Generating Availability Data System. North American Electric Reliability Corporation. July 2011. pp. 7, 17. Retrieved 13 March 2014.
2. "Security models for web-based applications" by James B. D. Joshi, Walid G. Aref, Arif Ghaffoor, Eugene H. Spafford and published in: Magazine Communications of the ACM CACM Homepage archive Volume 44 Issue 2, Feb. 2011 Pages 38-44 ACM New York, NY, USA <https://dl.acm.org/citation.cfm?id=359224>
3. "System And Method For Managing User Authentication And Service Authorization To Achieve Single-Sign-On To Access Multiple Network Interfaces" created by inventors Pei Chia and Hong Cheng for Panasonic Corp in 2004
4. <https://patents.google.com/patent/US20080072301A1/en>
5. "A survey of Web security" by A.D. Rubin ; D.E. Geer becomes a part of IEEE standards first published in Published in: Computer (Volume: 31, Issue: 9, Sept. 1998) pages: 34 – 41, but republishing is going till nowadays [<http://ieeexplore.ieee.org/abstract/document/708448/?reload=true>]
6. "Methods and systems for providing secure access to a hosted service via a client application" by Anthony J. Yeates, Pavel A. Dournov, Sumeet Updesh Shrivastava, Shankar Arunachalam Bharadwaj, Donna L. Whitlock [<https://patents.google.com/patent/US9858562B2/en>]
7. "Methods and systems for creating and managing network groups" by Thomas M. Kludy, Ashish Gujarathi, Ricardo Fernando Feijoo
8. <https://patents.google.com/patent/US9906461B2/en>

УДК 004.05(075)

Бортник К.Я., Ломінська Г.Ю.

Луцький національний технічний університет

ТЕХНОЛОГІЇ АНАЛІЗУ НАСЛІДКІВ КІБЕРАТАК

Бортник К.Я., Ломінська Г.Ю. Технології аналізу наслідків кібератак. У статті розкриваються особливості та характеристики кібербезпеки і розглядаються способи, якими фахівці з кібербезпеки аналізують наслідки кібератак. Пояснюються різні категорії вразливостей програмного та апаратного забезпечення та систем безпеки.

Ключові слова: кібербезпека, кіберзлочинність, вразливість.

Бортник К.Я., Ломинский Г.Ю. Технологии анализа последствий кибератак. В статье раскрываются особенности и характеристики кибербезопасности и рассматриваются способы, которыми специалисты по кибербезопасности анализируют последствия кибератак. Объясняются различные категории уязвимостей программного и аппаратного обеспечения и систем безопасности.

Ключевые слова: кибербезопасность, киберпреступность, уязвимость.

Bortnyk K.Ya., Lominskaya G.Yu. Technologies for analyzing the effects of cyber attacks. The article reveals the features and characteristics of cybersecurity and examines the ways in which cybersecurity experts analyze the effects of cyber attacks. Explains the various vulnerabilities of software and hardware and security systems.

Key words: cyber security, cybercrime, vulnerability.

Актуальність проблеми. За оцінками експертів правоохоронних органів країн Центральної та Східної Європи з питань боротьби з комп'ютерною злочинністю, прибутки злочинців від злочинів у сфері використання електронно-обчислювальних машин посідають третє місце після доходів наркоторговців і від продажу зброї.

Одним із досягнень сучасної людини, яка невинно розвивається в галузі ІТ-технологій стало впровадження в процес управління та інші галузі життя суспільства електронно-обчислювальної техніки, без якої зберігання, обробка та використання дуже великої кількості інформації було б неможливим. Під'єднання до електронної інформаційної мережі стало невід'ємною частиною нашого повсякденного життя. Усі види організацій, такі як медичні, фінансові та навчальні заклади, використовують цю мережу для ефективного функціонування. Вони використовують мережу для збору, обробки, зберігання та обміну великою кількістю цифрової інформації. Оскільки велика кількість цифрових даних збирається та спільно використовується, захист цієї інформації стає ще більш важливим для нашої національної безпеки та економічної стабільності.

У розвиток науки, техніки та інших сфер знань це досягнення принесло неоціненну користь. Однак вигоди, котрі можна отримати завдяки використанню такої техніки, стали використовуватись і в злочинних цілях. Так і з'явився новий вид неправомірної діяльності – комп'ютерні злочини, наслідки яких навіть не йдуть у порівняння зі шкодою від інших злочинів. За оцінками експертів правоохоронних органів країн Центральної та Східної Європи з питань боротьби з комп'ютерною злочинністю, прибутки злочинців від злочинів у сфері використання електронно-обчислювальних машин посідають третє місце після доходів наркоторговців і від продажу зброї, а завдані збитки вже зараз оцінюються мільярдами доларів. Тільки в США щорічно економічні збитки від такого роду злочинів становлять близько ста мільярдів доларів. До середини вісімдесятих років минулого століття у Великобританії збитки від комп'ютерних злочинів становить 750 мільйонів фунтів стерлінгів. В даний час вони зросли вдвічі. Тому питання безпеки – це питання номер один.

Кібербезпека - це постійні зусилля спрямовані на захист мережевих систем та всіх даних від несанкціонованого використання або заподіяння шкоди. На особистому рівні вам потрібно захистити вашу ідентичність, ваші дані та ваші електронні пристрої. На корпоративному рівні відповідальність кожного полягає в захисті репутації організації, даних та клієнтів. На державному рівні це національна безпека, а також на карту поставлено безпеку та добробут громадян.

Конфіденційність, цілісність та доступність, відома як триада CIA, є керівним принципом для безпеки інформації для організації. Конфіденційність - гарантує конфіденційність даних, обмежуючи доступ до них через механізм аутентифікації. Цілісність гарантує точність і достовірність інформації. Доступність гарантує, що інформація доступна для авторизованих користувачів.

Отже, тепер власне про комп'ютерні злочини. Їх умовно можна розділити на два типи: злочини, що вчиняються прямим втручанням у роботу комп'ютера, і злочини, в яких комп'ютер є необхідним технічним засобом для їх вчинення.

Наведемо до прикладу види злочинів, що пов'язані з втручанням у роботу ПК.

1. Несанкціонований доступ до інформації, що зберігається в комп'ютері.

Такий доступ здійснюється, зазвичай, з використанням несправжнього імені користувача, зміною MAC-адреси пристрою, використанням інформації, що залишилася в історії комп'ютера після виконання деяких операцій, модифікацією програмного та інформаційного забезпечення, розкраданням носія інформації, встановленням засобів запису та копіювання, що підключаються до каналів передачі даних.

Людей, які здійснюють несанкціонований доступ в чужі комп'ютерні та інформаційні мережі називають "електронними корсарами", "комп'ютерними піратами" або ж просто хакерами.

Несанкціонований доступ до файлів законного користувача здійснюється пошуком слабких місць в захисті системи. Одного разу виявивши їх, порушник може дослідити інформацію, що міститься в системі, копіювати її, повертатися до неї багато разів, подібно до того, як покупець розглядає товари на вітрині.

Програмісти іноді допускаються помилок в програмах, які не вдається виявити в процесі налагодження. Автори великих складних програм можуть не помітити деяких елементарних помилок. Уразливі місця іноді виявляються і в електронних ланцюгах. Всі ці недбалості, помилки призводять до появи "проломів".

Зазвичай вони все-таки виявляються при перевірці, редагуванні, налагодженні програми, але абсолютно позбавитися від них неможливо.

2. Введення в програмне забезпечення "логічних бомб", які спрацьовують при виконанні певних умов і лише частково або повністю виводять з ладу комп'ютерну систему.

"Тимчасова бомба" - різновид "логічної бомби", яка спрацьовує у визначений хакером час.

Такий так званий "троянський кінь", тобто вірус, полягає в таємному введенні в чужу програму таких команд, що дозволяють здійснювати нові, не заплановані власником програми, але вигідні хакеру функції і одночасно зберігати колишню працездатність.

За допомогою "троянського коня" злочинці, наприклад, можуть відрхувати на свій рахунок певну суму з чужого рахунку.

Програмні коди серйозних організацій зазвичай надзвичайно складні. Вони складаються з сотень, тисяч, а іноді і мільйонів команд. Тому "троянський кінь" з кількох десятків команд навряд чи може бути виявлений, якщо, звичайно, немає підозр щодо цього. Але і в останньому випадку експертам-програмістам потрібно багато часу, щоб знайти його.

У США набула поширення форма комп'ютерного вандалізму, при якій "троянський кінь" руйнує через якийсь проміжок часу всі програми, що зберігаються в пам'яті ПК. У багатьох комп'ютерах, що надходять в продаж, виявляють "тимчасову бомбу", яка "вибухає" у найнесподіваніший момент, руйнуючи всю бібліотеку даних. Не слід думати, що "логічні бомби" - це екзотика, невластива нашому суспільству.

Розробка та розповсюдження комп'ютерних вірусів.

"Троянським конем" зазвичай називають комп'ютерні віруси. Вони, вивівши на робочий стіл текст типу: "зітри всі дані цієї програми, перейди у наступну і зроби те ж саме", мають можливість переходити через комунікаційні мережі з однієї системи в іншу, поширюючись як вірусне захворювання.

Виявляється вірус не відразу: перший час комп'ютер зберігає його у собі як звичайну інформацію чи програму, оскільки для маскування вірус нерідко використовується в комбінації з "логічною бомбою" або "тимчасовою бомбою". Вірус спостерігає за всією оброблюваною інформацією і може переміщатися, використовуючи пересилання цієї інформації.

Починаючи діяти (перехоплювати управління), вірус дає команду комп'ютеру, щоб той записав заражену версію програми. Після цього він повертає програмі управління. Користувач нічого не помітить, тому що його комп'ютер знаходиться в стані "здорового носія вірусу". Виявити цей вірус можна, тільки володіючи надзвичайно розвиненою інтуїцією програміста, оскільки ніякі порушення в роботі ЕОМ в даний момент не видають себе. А в один прекрасний день комп'ютер виходить з ладу. Завдяки таким махінаціям, хакери мають змогу заробити не законним шляхом. Експертами зібрано досє листів від шантажистів, які вимагають перерахування великих сум

грошей в одне з відділень американської фірми "ПК Cyborg"; у разі відмови злочинці погрожують вивести комп'ютери з ладу. Для пошуку і виявлення зловмисників створені спеціальні загони англійських детективів. За оцінкою фахівців в "зверненні" знаходиться більше 100 типів вірусів.

Але всі їх можна розділити на два різновиди, виявлення яких різне за складністю – це "вульгарний вірус" і "роздроблений вірус". Програма "вульгарного вірусу" написана єдиним блоком, і при виникненні підозр у зараженні ЕОМ експерти можуть виявити її на початку поширення. Ця операція вимагає вкрай ретельного аналізу всіх складових операційної системи ЕОМ. Програма "роздробленого вірусу" розділена на частини, на перший погляд, не має між собою зв'язку. Ці частини містять інструкції, які вказують комп'ютеру, як зібрати їх воєдино щоб відтворити і, отже, розмножити вірус. Таким чином, він майже весь час знаходиться в "розподіленому" стані, лише на короткий час своєї роботи збирається у єдине ціле. Як правило творці вірусу вказують йому число репродукцій, після досягнення якого він стає агресивним.

Віруси можуть бути впроваджені в операційну систему, прикладну програму або в мережевий драйвер.

Види вірусів залежать від цілей, переслідуваних їх творцем. Ознаки їх можуть бути відносно доброякісними, наприклад, уповільнення у виконанні програм або поява світлої точки на екрані дисплея (т. зв. "Італійський стрибунець"). Ознаки можуть бути еволютивними, і "хвороба" буде загострюватися в міру своєї течії. Так, з незрозумілих причин, програми починають переповнювати магнітні диски, в результаті чого істотно збільшується обсяг програмних файлів. Нарешті, ці прояви можуть бути катастрофічними і призвести до стирання файлів і знищення програмного забезпечення.

Мабуть, у майбутньому будуть з'являтися принципово нові види вірусів. Наприклад, можна собі уявити (поки подібних повідомлень не було) свого роду "троянського коня" вірусного типу в електронних ланцюгах. Справді, поки мова йде тільки про зараження комп'ютерів. А чому б - не мікросхем? Адже вони стають все більш потужними і перетворюються на подобу ЕОМ. І їх необхідно програмувати. Звичайно, ніщо не може безпосередньо "заразити" мікросхему. Але ж можна заразити комп'ютер, який використовується як програматор для тисячі мікросхем.

Способи розповсюдження комп'ютерних вірусів ґрунтуються на здатності вірусу використовувати будь-який носій даних, що передаються як "засіб пересування". Тобто з початку зараження є небезпека, що ЕОМ може створити велику кількість засобів пересування і в наступні години вся сукупність файлів і програмних засобів буде заражена. Таким чином, дискета або магнітна стрічка, перенесені на інші ЕОМ, здатні заразити їх. І навпаки, коли "здорова" дискета вводиться в заражений комп'ютер, вона може стати носієм вірусу. Зручними для поширення великих епідемій виявляються телекомунікаційні мережі. Досить одного контакту, щоб персональний комп'ютер був заражений або заразив той, з яким контактував. Однак найпоширеніший спосіб зараження - це копіювання програм, що є звичайною практикою у користувачів персональних ЕОМ. Так скопійованими виявляються і заражені програми.

Фахівці застерігають від копіювання крадених програм. Однак, іноді і офіційно представлені програми можуть бути джерелом зараження.

Природно, що проти вірусів були прийняті надзвичайні заходи, що призвели до створення текстових програм-антивірусів. Захисні програми поділяються на три види: фільтруючі (що перешкоджають проникненню вірусу), проти інфекційні (постійно контролюють процеси в системі) і противірусні (налаштовані на виявлення окремих вірусів).

Однак розвиток цих програм поки що не встигає за розвитком комп'ютерної епідемії.

Зауважимо, що побажання обмежити використання неперевіреного програмного забезпечення швидше за все так і залишиться практично нездійсненим. Це пов'язано з тим, що фірмові програми на "стерильних" носіях коштують чимало у валюті. Тому уникнути їх неконтрольованого копіювання майже неможливо.

Заради справедливості слід зазначити, що поширення комп'ютерних вірусів має і деякі позитивні сторони. Зокрема, вони є, мабуть, кращим захистом від викрадачів програмного забезпечення. Найчастіше розробники свідомо заражають свої дискети яких-небудь нешкідливим вірусом, який добре виявляється будь-яким антивірусним тестом. Це служить досить надійною гарантією, що ніхто не ризикне копіювати таку дискету.

Незалежно від того, яким типом зловмисного ПЗ інфікована система, можна виділити найпоширеніші симптоми того, що пристрій заражено:

- Збільшується навантаження на центральний процесор;
- Знижується швидкодія комп'ютера;
- Комп'ютер часто зависає або дає збої;
- Знижується швидкість перегляду веб-сторінок;
- З'являються незрозумілі проблеми з мережними з'єднаннями;
- Модифікуються файли;
- Видаляються файли;
- Наявність невідомих файлів, програм або значків на робочому столі;
- Запущено невідомі процеси;
- Програми самостійно знімаються з виконання або переконфігуровуються;
- Електронна пошта надсилається без відома та згоди користувача.

Попередження комп'ютерних злочинів.

При розробці комп'ютерних систем, вихід з ладу або помилки в роботі яких можуть призвести до тяжких наслідків, питання комп'ютерної безпеки стають першочерговими. Відомо багато заходів, спрямованих на попередження злочину. Виділимо з них технічні, організаційні та правові.

До технічних заходів можна віднести захист від несанкціонованого доступу до системи, резервування особливо важливих комп'ютерних підсистем, організацію обчислювальних мереж з можливістю перерозподілу ресурсів у разі порушення працездатності окремих ланок, прийняття конструкційних заходів захисту від розкрадань.

До організаційних заходів віднесемо охорону обчислювального центру, ретельний підбір персоналу, виключення випадків ведення особливо важливих робіт лише однією людиною, наявність плану відновлення працездатності центру після виходу його з ладу, організацію обслуговування обчислювального центру сторонньою організацією або особами, незацікавленими в приховуванні фактів порушення роботи центру, універсальність засобів захисту від усіх користувачів (включаючи вище керівництво), покладання відповідальності на осіб, які повинні забезпечити безпеку центру, вибір місця розташування центру і т.п.

До правових заходів слід віднести розробку норм, що встановлюють відповідальність за комп'ютерні злочини, захист авторських прав програмістів, вдосконалення кримінального і цивільного законодавства, а також судочинства. До правових заходів належать також питання громадського контролю за розробниками комп'ютерних систем і прийняття міжнародних договорів про їх обмеження, якщо вони впливають або можуть вплинути на військові, економічні та соціальні аспекти життя країн, що укладають угоду.

Кінець ери відповідності нормам і стандартам в кіберпросторі

У найближчі роки, скоріш за все, тенденція відходу від норм і стандартів під час боротьби з атаками в кіберпросторі буде посилюватись на користь більш творчих рішень. Робота спеціалістів з кібербезпеки буде зосереджена на оцінці реальних загроз і визначення пріоритетності своїх дій, що в свою чергу вимагатиме від них глибокого розуміння проблем, творчого підходу та досить високої експлуатаційної гнучкості.

Висновки. Підсумовуючи усе вище сказане, можна зазначити, що основними тенденціями розвитку ситуації в області кібербезпеки буде поява все більш складних методів кібератак на елементи критичної інфраструктури, посилення інформаційної боротьби в кіберпросторі, зростання загроз для соцмереж, втручання у функціонування бізнесу та проведення фінансових операцій. Наслідком цих тенденцій буде збільшення рівня контролю за користувачами мережі Інтернет, посилення регулювання на національному та міжнародному рівнях діяльності в кіберпросторі, активізація дискусії стосовно збільшення фінансування та інновацій в сектор кібербезпеки, а також покращення фахової підготовки спеціалістів з кібербезпеки.

1. Мастяниця Й.І., Соснін О.В., Шиманський Л.Є. Захист інформаційних ресурсів України: проблеми і шляхи їх розв'язання. - К.: Національний інститут стратегічних досліджень, 2000. - 98 с.
2. П. Браїловський М.М., Головень СМ. та інші. - Технічний захист інформації на об'єктах інформаційної діяльності/ За ред. Проф. В.О. Хорошка. -К.: ДУІКТ, 2007. - 178с
3. <https://1391723.netacad.com/courses/623568> Академія Cisco України
4. Kochergin AN Information and spheres of its manifestation: monograph. - Golitsyno: GPI of the Federal Security Service of the Russian Federation, 2008. - 272 p.

УДК 004,254 (045)

Бортник К.Я., Ольшевський О.В., Пашук В.Ю.
Луцький національний технічний університет

ІНТЕРНЕТ РЕЧЕЙ ТА ЯК ВІН ЗМІНИТЬ НАШЕ ЖИТТЯ У МАЙБУТНЬОМУ

Бортник К.Я., Ольшевський О.В., Пашук В.Ю. Інтернет речей та як він змінить наше життя у майбутньому. В статті пояснюється що таке «інтернет речей», принципи його функціонування та висувуються припущення стосовно подальшого розвитку цієї концепції. Показано авторське бачення впливу цієї ідеї на майбутнє життя людства.

Ключові слова: інтернет речей, мережі, технології, розумний дім, штучний інтелект

Bortnyk K.Ya., Olshevsky O.V., Pashchuk V.Y. Internet of things and how it will change our life in the future.

The article explains what «Internet of things» are, the principles of functioning, and speculates about further development of this concept. The author's opinion of the influence of this idea on the future life of mankind is shown.

Key words: internet of things, networks, technology, artificial intelligence, smart house.

Бортник Е.Я., Ольшевский А.В., Пашук В.Ю. Интернет вещей и как он изменит нашу жизнь в будущем.

В статье объясняется что такое «интернет вещей», принципы его функционирования и выдвигаются предположения относительно дальнейшего развития этой концепции. Показано авторское видение влияния этой идеи на будущую жизнь человечества.

Ключевые слова: интернет вещей, сети, технологии, умный дом, искусственный интеллект

Постановка наукової проблеми. З кожним роком кількість «розумних пристроїв», що їх винаходять люди невпинно збільшується. Розумнішають звичні нам речі, наприклад годинники, кавові машинки, холодильники, дверні замки, вікна, освітлювальні прилади та інші. Все це стало можливим завдяки стрімкому розвитку технологій. Сьогодні кожен з нас має такі обчислювальні потужності у кишені, про які ще 50 років тому люди могли лише мріяти. Розумні речі об'єднуються одне з одним у мережі. Давайте уявимо, що ви живете у розумному будинку. Ви встаєте зранку і браслет, що знаходиться на вашій руці сповіщає кавоварку, що ви прокинулися. Вона, у свою чергу, заварює каву і перевіряє її запас; якщо він закінчується, кавоварка сповіщає про це кур'єра. Аналогічну процедуру виконує і холодильник. Отримавши цю інформацію кур'єр аналізує де найкраще виконати покупки і поновлює запаси. Таким чином, коли ви прокинетесь, кава вже буде зварена, а холодильник - повний продуктів. Це лише один приклад використання цієї концепції з мільйонів можливих. Мережа таких будинків складатиме квартал, який зможе виконувати самоаналіз. Так утвориться розумне місто і т.д.. Перші кроки до цього робляться ще з початку 2000-х років. Вже існує концепція такого будинку, а окремі елементи цієї системи успішно функціонують у нашому повсякденному житті.

Аналіз досліджень. Концепцію «інтернету речей» досліджували такі дослідники Герман Копетц, Кевін Ештон, Антоніо Лера, Даніель Міоранді та інші у своїх статтях. Починаючи з 2009 року у Брюсселі проходять конференції присвячені «Інтернету речей», на яких виступають науковці та лідери провідних ІТ-компаній.

Метою нашої статті є пояснення принципу роботи «інтернету речей», донесення його сутності до широкого кола аудиторії та залучення читачів до роздумів стосовно майбутнього, розвинутого під впливом цієї технології.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Вперше про «Інтернет речей» заговорив Кевін Ештон у 1999р. щоб пояснити мережу в якій би взаємодіяли між собою різні об'єкти з навколишнього середовища завдяки сенсорам та інтернету.

По суті, «Інтернет речей» це мережа пристроїв, кожен з яких під'єднаний до всесвітньої павутини та може взаємодіяти з іншими пристроями, що входять до цієї мережі. Ця взаємодія реалізовується завдяки системі датчиків, трансляторів та сенсорів для збору, передачі та обробки інформації. Для утворення такої мережі пристроїв необхідно, щоб виконувалися наступні умови:

- Кожному пристрою має надаватися свій ідентифікатор. Це може бути досягнуто завдяки використанню QR-кодів, GPS, радіочастотної ідентифікації та інших технологій.
- Повинен відбуватися збір даних про зміни у навколишньому середовищі та зміни самого пристрою.
- Інформація, що надходить до пристрою, має оброблятися інтегрованим комп'ютером.

- Пристрої однієї мережі повинні обмінюватися між собою даними. Для цього найчастіше використовують Wi-Fi та Bluetooth.

Якщо ми хочемо об'єднати не тільки прилади у своєму будинку, то необхідно скористатися глобальними методами зв'язку. Сьогодні основою для цього по всьому світі є Інтернет. Саме завдяки його стрімкому розвитку і поширенню, ідея «Інтернету речей» вже не здається такою туманною. На даний момент найпоширенішим мережевим протоколом є IPv4. Він дозволяє використовувати до 4.3 мільярда унікальних адрес. Цього вже не достатньо, щоб одночасно надати адресу в інтернеті кожному пристрою, який здатен мати до нього доступ. Тому найближчим часом популярності набере стандарт IPv6, який забезпечить унікальними адресами не менше 300 мільйонів пристроїв на одного користувача. Саме він стане основою для впровадження «Інтернету речей» у майбутньому в межах усієї планети.

Перспективи та небезпеки розвитку «Інтернету речей». В цей час тренд «Інтернету речей» набирає все більшу популярність, що не може не відобразитися на його розвитку. В 2017 році кількість приладів підключених до Інтернету, за даними Statista, перевищили 20 мільярдів. А до 2020 року їхня кількість може зрости до 50 мільярдів, передбикає Cisco(рисунок 1).

Програми для підключення великої кількості девайсів до мережі є досить обширними. Для зручності було запропоновано декілька категорій їх розподілу, більшість з яких узгоджуються між додатками споживачів, підприємств та інфраструктур. Джордж Осборн, колишній британський Канцлер скарбниці, стверджував, що "Інтернет речей" - це наступний етап інформаційної революції.

Можливість мережевого вбудованого пристрою з обмеженим центральним процесором, пам'яттю та ресурсами живлення означає, що "Інтернет речей" може застосовуватися практично у всіх галузях нашої життєдіяльності.

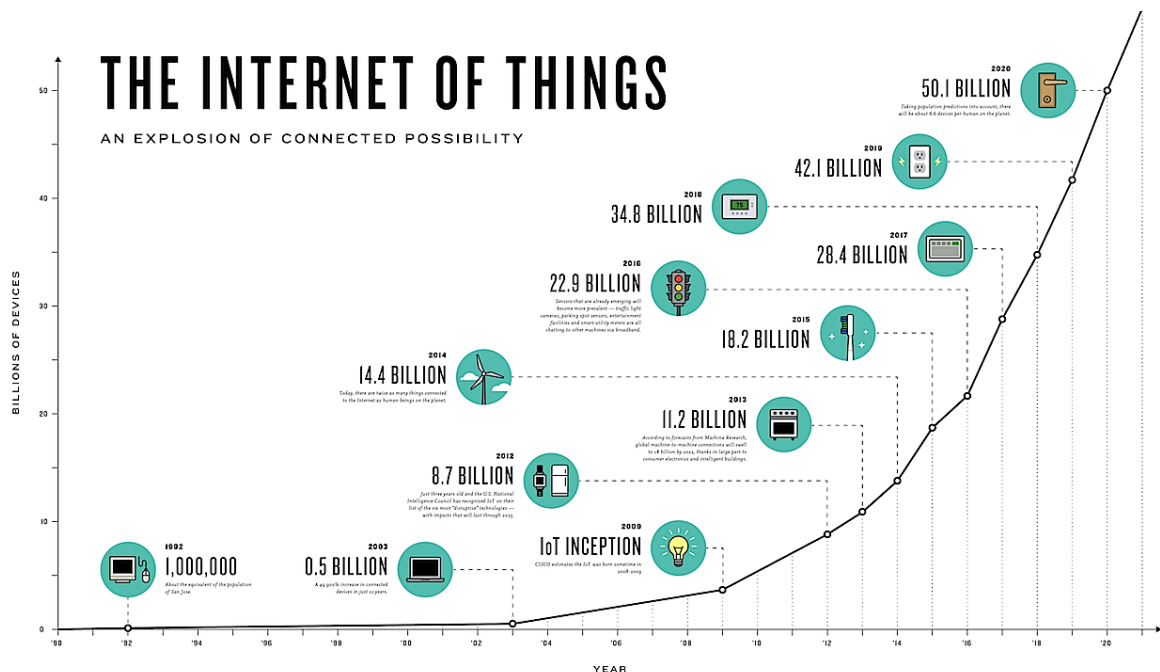


Рис.1

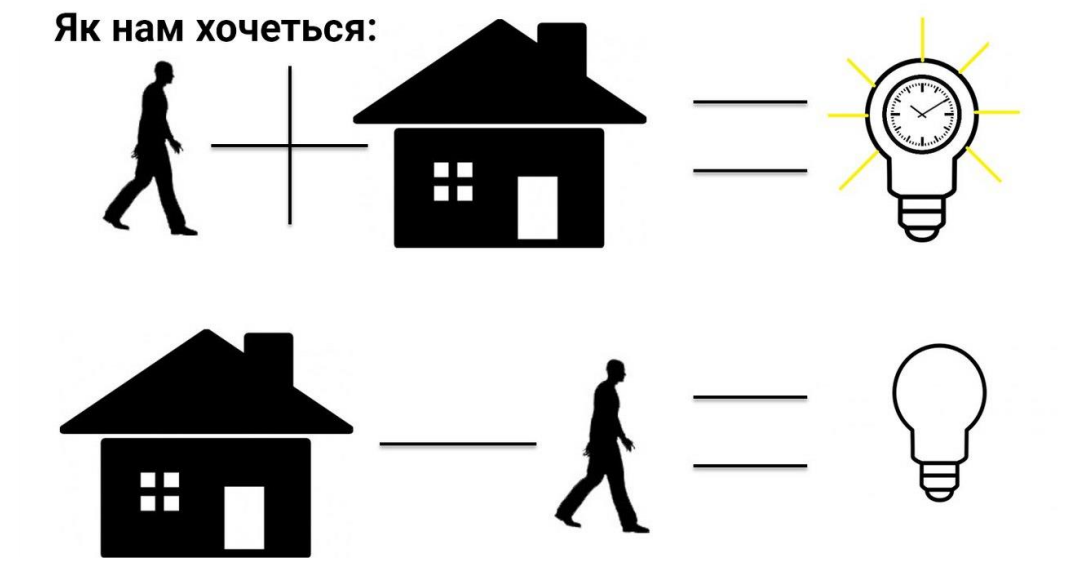


Рис.2

Такі системи могли б відповідати за систематизований збір інформації в установках, починаючи від природних екосистем закінчуючи різноманітними фабриками та будинками, тобто приймати участь в сферах екологічного спостереження та містобудування.

Інтелектуальні торгові системи, наприклад, можуть відстежувати певні споживачські звички покупців у магазині шляхом відстеження різноманітних сенсорів. Тоді ці користувачі зможуть отримувати спеціальні пропозиції щодо своїх улюблених продуктів або навіть розташування необхідних для них предметів, які їхній холодильник автоматично передає на телефон. Додаткові приклади спостереження, аналізу та співпраці цих девайсів можна спостерігати у програмах, які мають справу з опаленням, постачанням води, електроенергії та енергоменеджменту, а також з системами транспортування, прикладом є рисунок 2. Інші програми, які може надати "Інтернет речей", - це розширення можливостей домашньої безпеки та домашньої автоматизації.

«Інтернет речей» це не тільки розумний холодильник чи кавоварка, яка сама починає готувати вам каву відразу як ви прокинулися. Його можна використовувати у будь-якій галузі, де щось можна автоматизувати. Особливо активно «Інтернет речей» розвивається в аграрному секторі, логістиці, концепції Smart City. Тобто там, де є потреба моніторингу стану об'єктів або в зборі великих даних з метою їх подальшого аналізу. «Інтернет речей» дає можливість економити на обслуговуванні обладнання: датчики збирають інформацію про його стан, тому техобслуговування і ремонт здійснюються саме тоді, коли це є необхідним. Профілактика - завжди дешевше ремонту. Наприклад, у власному будинку датчики самі будуть знімати покази лічильників, визначати скільки енергії споживає кожна лампочка чи будь-який прилад підключений до мережі, температура у будинку буде встановлюватися автоматично відповідно до ваших потреб. Світло, вода, газ – все це буде автоматично вимикатися відразу після того, як ви покинете будинок. Тобто мова йде не про «розумний» будинок, адже в такому випадку людина все ще повинна контролювати всі процеси у будинку, а більше про «автономний» будинок, де виконання процесів кожного приладу будуть базуватися на аналізі потреб користувача (людини).

Датчики переміщення громадського транспорту, планування маршрутів на основі переміщення жителів, датчики наповнення сміттєвих баків, відеоспостереження, датчики шуму та забруднення зроблять життя у містах більш безпечним та зручним. А результати цього глобального аналізу мережа може передавати до влади міста для кращого розуміння потреб його жителів.

У аграрній сфері «Інтернет речей» дозволить спростити агрономам догляд за своїми плантаціями. Наприклад, датчики в землі будуть фіксувати показники вологості ґрунту, потреби

рослин у добривах. Дрони будуть вести запис даних з неба і на основі цих даних нейронні мережі можуть приймати рішення щодо виниклих проблем чи потреб.

Завдяки «Інтернету речей» доставка будь-яких товарів з виробництва або зі складів до магазинів більш передбачувана - що важливо як для кінцевого споживача, так і для бізнесу. Транспортні компанії можуть відстежити, де перебуває автомобіль, або в який момент йому потрібно під'їжджати на завантаження. Датчики сканують стан і місцезнаходження автомобіля, що, у свою чергу, позбавляє власників логістичних компаній від зайвих дзвінків і витрат на ремонт.

Хороший приклад накопичення даних і подальшого їх використання в різних додатках – фітнес-платформи Apple HealthKit і Google Fit. Вони збирають інформацію з фітнес-трекерів, смарт-браслетів, «розумних» годинників, обробляють - і видають загальну картину про стан здоров'я, фізичні навантаження і можливі проблеми з тиском, серцебиттям, рівнем цукру в крові, перепадами температури тіла. А якщо додати синхронізацію цього аналізу з мережею лікарень, то вдасться управляти станом та якістю здоров'я людей.

Не зважаючи на велику користь та переваги розвитку «Інтернету речей», є і певні небезпеки у його використанні. Головним ризиком у використанні - це питання безпеки. Оскільки все більше і більше пристроїв підключаються до мережі, вони вносять вклад в категорію пристроїв, які легко захоплюються в ботнети і можуть бути використані для розподілених нападів. Використання розподілених атак робить відстеження джерел атаки більш складним процесом, в той же час полегшуючи можливість злову пристроїв і додатків, на які орієнтовані атаки.

Фізичні атаки можуть пошкодити датчики пристроїв «Інтернету речей» або навіть привести їх у повністю непрацездатний стан, що являє собою явну загрозу безпеці. Наприклад, зловмисник може увійти в будинок, де розташований датчик, і виявити супутні електронні та фізичні сигнали інших сенсорів за допомогою обладнання для виявлення радіо-, тепло-, магнітних, візуальних та інших електронних сигналів. Потім зловмисник може визначити розташування датчиків на підставі властивостей сигналів, після чого вони можуть бути відключені фізично, знищені або вкрадені. Фізичне руйнування може бути здійснено з використанням нагрівання, фізичної сили або порушення цілісності ланцюга датчиків. Крім того, легко запустити фізичні атаки з використанням старих технологій через уразливість датчиків, особливо невеликих. Атаки такого роду неминучі для сенсорних мереж «Інтернету речей». Оскільки зловмисник знаходиться в безпосередній близькості до мережі при атаці такого роду, у нього є можливість реагувати на захисні механізми, на відміну від віддалених атак.

Також є очевидним, що якщо зловмисник отримує доступ до певної інфраструктури, то він може відновити інформацію, що проходить через неї. Поки заходи безпеки спрямовані на захист даних та інформації, ймовірність того, що зловмисник зможе отримати доступ до самої системи і вкрати дані, реальна.

Однієї з основних проблем для «Інтернету речей», якщо говорити не про домашнє застосування, а про бізнес або державу, є ресурси для зберігання і обробки інформації. Власна інфраструктура вимагає великих інвестицій відразу ж, причому будувати її доведеться «про запас», частина обладнання буде банально простоювати.

Через це, в еру «Інтернету речей», багато компаній звернуть свою увагу на хмарні сховища. Вони вимагають мінімум часу на розгортання, дозволяють купувати рівно стільки ресурсів, скільки необхідно компанії на даний момент. Хмари - це практично необмежений ресурс для збору і аналізу великих даних. Аналіз дозволяє знаходити приховані закономірності і є джерелом інсайтів, які допомагають бізнесу (або державі, якщо мова йде, наприклад, про дані міст) розвиватися, оптимізувати витрати, знаходити потенційні загрози і слабкі місця, вирішувати несподівані проблеми. Хоча однією з проблем даного способу збереження інформації є те, що власник цих даних не знає, де вони можуть знаходитися і хто має доступ до них.

Тому, ще однією вадою у роботі «Інтернету речей» є неможливість впевнено сказати, що за нами не будуть стежити через наші ж «розумні системи». Хоча ми не можемо цього і зараз стверджувати, глобальне і тотальне спостереження однозначно вплине на наше життя. Доки

концепція «Інтернету речей» буде базуватися на «чесному слові», ніщо не завадить людській природі взяти своє і використати дані чужий користувачів для своєї вигоди.

Хоча для вирішення питання про безпеку даних в мережі «Інтернету речей» сьогодні використовують блокчейн. Його застосовують в «Інтернеті речей» для безпеки та цілісності персональних даних кожного користувача. Сам блокчейн являє собою розподілену базу даних, яка доступна кожному. На його основі можуть бути здійснені різноманітні юридичні й фінансові операції, тобто оплата криптовалютами. У цей час блокчейн не можливо зламати через не здатність комп'ютерів до обчислень такого масштабу. Хоча з появою квантових комп'ютерів блокчейн перестане бути таким надійним.

Проте на даний час не вистачає стандартизації і для підключення та використання одного «розумного» приладу потрібна одна технологія, а для іншого – друга. Зараз вважають, що найкращим стандартом зв'язку в «Інтернеті речей» буде технологія 5G мережі. Але на реальну продукцію на основі цієї технології прийдеться зачекати ще декілька років.

Висновки. Хоча «Інтернет речей» як технологія для об'єднання в одну мережу пристроїв, що нас оточують, почав розвиватися ще у середині першого десятиріччя 20 століття, він не є таким далеким майбутнім, як нам здається. Разом з тим, це не означає, що люди готові до цього майбутнього. Будь – яка технологія може бути використана не так, як це задумувалося її творцем і «Інтернет речей» не є виключенням. Саме це надбання людства, за умови глобального поширення, може стати одним із тих, що змінить світовий порядок. Разом з перспективами, що в собі несе ця концепція існують і небезпеки її впровадження. Так, поки ми не зможемо гарантувати безпеку інформації, що буде збиратися такими «розумними системами» та їхню стійкість до кібератак, варто не квапитися робити все навколо «розумним», адже наслідки можуть бути непередбачуваними.

1. Лекторій. Що таке інтернет речей і навіщо він потрібен? [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – «Видавничий дім «МЕДІА-ДК», 2014-2018. – Режим доступу: <https://nv.ua/ukr/science/lectures/lektorij-shcho-take-internet-rechey-i-navishcho-vin-potriben-1326653.html> (дата звернення 02.04.2018) – Назва з екрана.
2. Що таке «інтернет речей»? [Електронний ресурс] : [Веб-сайт]. – Електронні дані. - 2017. – Режим доступу: <http://thefuture.news/iot/> (дата звернення 02.04.2018) – Назва з екрана.
3. Інтернет речей. Новомодне захоплення чи технологія, що змінює світ? [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Економічна правда, 2005-2017. – Режим доступу: <https://www.epravda.com.ua/publications/2015/06/2/544801/> (дата звернення 02.04.2018) – Назва з екрана.
4. Інтернет речей [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – 2018. – Режим доступу: https://uk.wikipedia.org/wiki/Інтернет_речей (дата звернення 02.04.2018) – Назва з екрана.

УДК 551.568.85

Гненний А.П. студент, Гордієнко Ю.Г. професор д.т.н.
НТУ України «Київський політехнічний інститут ім. Ігоря Сікорського»

АКТУАЛЬНІ МЕХАНІЗМИ ІДЕНТИФІКАЦІЇ ОСОБИСТОСТІ В ІНТЕРНЕТІ РЕЧЕЙ

Гненний А.П., Гордієнко Ю.Г. Актуальні механізми ідентифікації особистості в інтернеті речей. Розкрито актуальні механізми ідентифікації особистості в Інтернеті речей. Наведено архітектуру IoT, яка включає чотири функціональних рівня. Запропоновано приклади веб-взаємодії з пристроями сенсорної мережі. Відокремлено 3 способи взаємодії з інтернет-речами: прямий доступ; доступ через шлюз; доступ через сервер. Оскільки локальні мережі інтернет-речей являють собою в основному бездротові сенсорні мережі, то шлюзи, використовувані в Інтернеті речей, аналогічні використовуваним в територіально-розподілених сенсорних мережах. Описано кілька способів організації шлюзів.

Ключові слова. Інтернет речей, мережа, бездротові стандарти, ідентифікація, сенсор.

Гненний А.П., Гордиенко Ю.Г. Актуальные механизмы идентификации личности в интернете вещей. Аннотация. Раскрыты актуальные механизмы идентификации личности в Интернете вещей. Приведена архитектура IoT, которая включает четыре функциональных уровня. Предложены примеры веб-взаимодействия с устройствами сенсорной сети. Обособленно 3 способа взаимодействия с интернет-вещами: прямой доступ; доступ через шлюз; доступ через сервер. Поскольку локальные сети интернет-вещей представляют собой в основном беспроводные сенсорные сети, шлюзы, используемые в Интернете вещей, аналогичные используемым в территориально-распределенных сенсорных сетях. Описано несколько способов организации шлюзов.

Ключевые слова. Интернет вещей, сеть, беспроводные стандарты, идентификация, сенсор.

Gnenny A.P., Gordienko Y. Actual mechanisms of personality identification on the Internet of things. Actual mechanisms of identity identification in the Internet of things are revealed. The architecture of the IoT, which includes four functional levels. Examples of web interaction with sensor network devices are offered. Isolation 3 way of interacting with the Internet of things: direct access; access through the gateway; access via server. Since the local networks of Internet of things are mostly wireless sensor networks, the gateways used in the Internet of things are similar to those used in geographically distributed sensor networks. Several ways to organize gateways are described.

Keyword. Internet of things, network, wireless standards, identification, sensor.

Вступ. Обмін даними в системах Інтернету речей зазвичай здійснюється за допомогою бездротових протоколів, придуманих десятки років назад. Саме бездротові стандарти обміну даними найкращим чином підходять для Інтернету речей. Пристрої Інтернету речей часто важко обслуговувати або вони зовсім знаходяться в недоступному місці. Очевидно, що вартість таких рішень не повинна бути високою, пристрої повинні бути енергоефективними, оскільки часто працюють в умовах відсутності промислової електричної мережі.

Різноманітність бездротових протоколів дивує, перерахуємо найбільш популярні з них. Bluetooth Low Energy – це стандарт, що активно розвивається, низького обміну даними в радіусі до 30 метрів, він енергоефективний і недорогий в реалізації. Wi-fi – високошвидкісний протокол обміну даними, застосовується для підключення пристрою до мережі Інтернет. Zigbee і Z-wave – два конкуруючих стандарти з комірчастою топологією (mesh), застосування яких дозволить з'єднати обладнання в єдину локальну мережу з точкою виходу в Інтернет, за вартістю реалізації – затратні та вимагають обов'язкової присутності концентратора для організації мережі. Крім розглянутих також широко застосовуються такі стандарти: 6LowPAN, Thread, NFC, Sigfox, а також стільниковий зв'язок [1].

Аналіз останніх досліджень та публікацій. Питання ідентифікації особистості в Інтернеті Речей досліджувалося на сторінках робіт багатьох вчених. До їх числа варто віднести: П.І. Семенченко [2], яка дослідила IoT платформи, без яких майже неможливо уявити собі правильну і ефективну роботу пристроїв Інтернету речей. Виділяються і описуються механізми роботи, основні функції платформ і конкретні приклади того, як з їх допомогою можна поліпшити роботу окремого пристрою або мережі пристроїв в цілому.

Шварц Марко [3] описав процес розробки недорогих, але ефективних пристроїв для Інтернету речей на основі популярного мікроконтролера з функцією Wi-Fi ESP8266. Розказано, як зчитувати, відправляти і відстежувати дані через хмарні сервіси і дистанційно керувати пристроями звідки завгодно, застосовувати ESP8266 для взаємодії з соціальними мережами Twitter і Facebook, відправляти користувачам ESP8266 повідомлення по email, SMS і push-каналах, організовувати міжмашинну взаємодію без участі людини, в тому числі через хмару. На практичних прикладах показано побудову простої системи домашньої автоматизації з керуванням

через хмару, а також розгортання власної хмарної платформи. Описано, як зробити дверний замок з керуванням через хмару, фізичний індикатор курсу валюти, бездротове садове обладнання та багато інших пристроїв на основі ESP8266.

У якості зарубіжних джерел варто відзначити роботи: Li S., Xu L., Wang X. [4], He W., Xu L. [5], Uckelmann D., Harrison M., Michahelles F. [6], Wang L., Xu L., Bi Z., Xu Y. [7].

Визначення мети та завдання дослідження. Метою даного дослідження виступає розкриття актуальних механізмів ідентифікації особистості в Інтернеті Речей.

Основна частина дослідження. Інтернет речей концептуально належить до мереж наступного покоління, тому його архітектура багато в чому схожа з відомою чотиришаровою архітектурою NGN. IoT складається з набору різних інфокомунікаційних технологій, що забезпечують функціонування Інтернету речей, і його архітектура показує, як ці технології пов'язані один з одним. Архітектура IoT включає чотири функціональних рівня (рис. 1), описаних нижче.

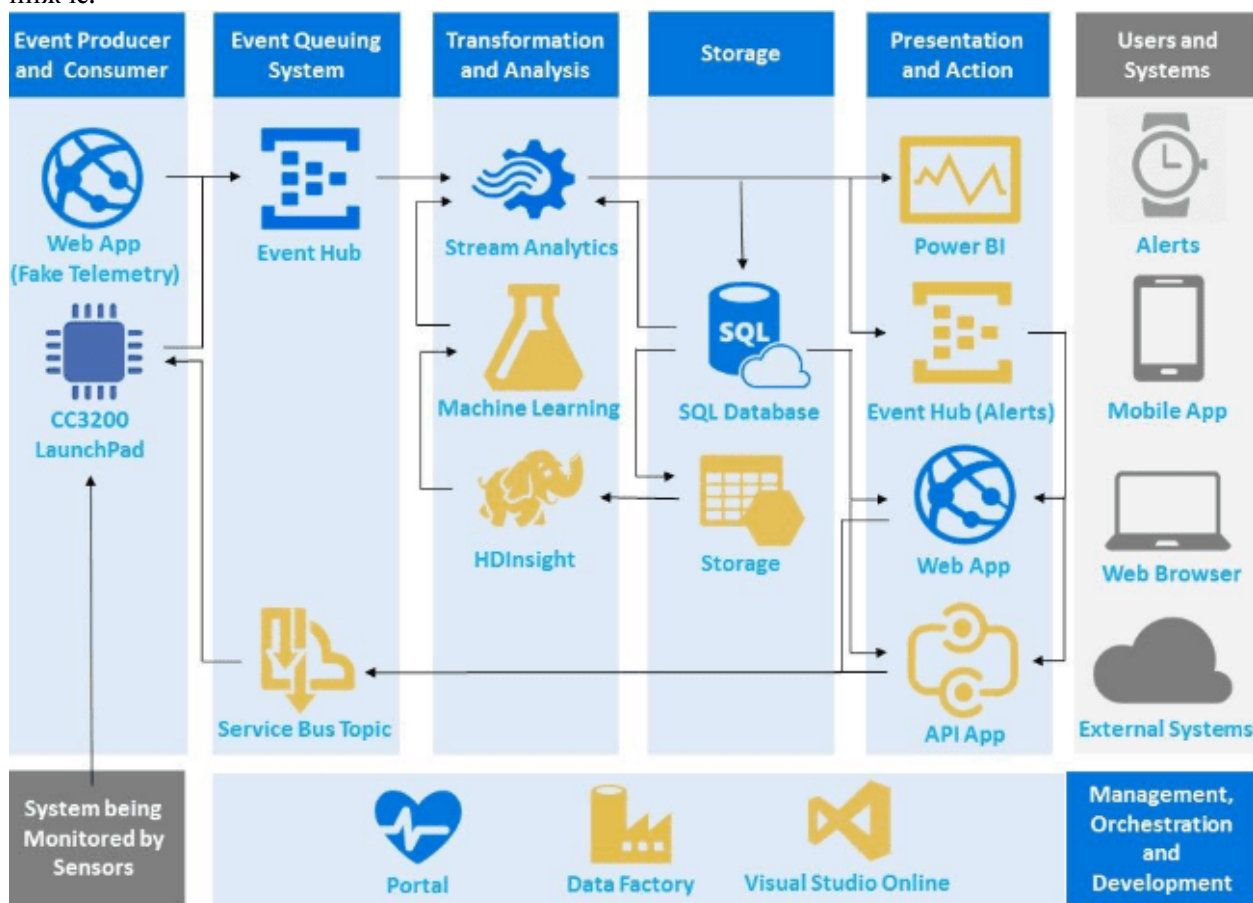


Рис. 1 – Архітектура IoT

1. Рівень сенсорів та сенсорних мереж.

Самий нижній рівень архітектури IoT складається з «розумних» (smart) об'єктів, інтегрованих з сенсорами (датчиками). Сенсори реалізують з'єднання фізичного та віртуального (цифрового) світів, забезпечуючи збір і обробку інформації в реальному масштабі часу. Мініатюризація, яка призвела до скорочення фізичних розмірів апаратних сенсорів, дозволила інтегрувати їх безпосередньо в об'єкти фізичного світу. Існують різні типи сенсорів для відповідних цілей, наприклад, для вимірювання температури, тиску, швидкості руху, місця розташування та ін. Сенсори можуть мати невелику пам'ять, даючи можливість записувати деяку кількість результатів вимірювань. Сенсор може вимірювати фізичні параметри контрольованого об'єкта/явища і перетворити їх в сигнал, який може бути прийнятий відповідним пристроєм. Сенсори класифікуються згідно з їх призначенням, наприклад, сенсори навколишнього середовища, сенсори для тіла, сенсори для побутової техніки, сенсори для транспортних засобів і т. д.

Більшість сенсорів потребує з'єднання з агрегатором сенсорів (шлюзом), які можуть бути реалізовані з використанням локальної обчислювальної мережі (LAN, Local Area Network), таких

як Ethernet і Wi-Fi або персональної мережі (PAN, Personal Area Network), таких як ZigBee, Bluetooth і ультраширокополосного бездротового зв'язку на малих відстанях (UWB, Ultra-Wide Band). Для сенсорів, які не вимагають підключення до агрегатора, їх зв'язок з серверами/додатками може надаватися використанням глобальних бездротових мереж WAN, таких як GSM, GPRS і LTE.

Сенсори, які характеризуються низьким енергоспоживанням і низькою швидкістю передачі даних, утворюють широко відомі бездротові сенсорні мережі (WSN, Wireless Sensor Network). WSN набирають все більшу популярність, оскільки вони можуть містити набагато більше сенсорів з підтримкою роботи від батарей і охоплюють великі площі.

2. Рівень шлюзів і мереж.

Великий обсяг даних, створюваних на першому рівні IoT численними мініатюрними сенсорами, вимагає надійної і високопродуктивної провідної або бездротової мережевої інфраструктури в якості транспортного середовища. Існуючі мережі зв'язку, що використовують різні протоколи, можуть бути використані для підтримки міжмашинних комунікацій M2M та їх додатків. Для реалізації широкого спектру послуг і додатків в IoT необхідно забезпечити спільну роботу безлічі мереж різних технологій і протоколів доступу в гетерогенній конфігурації. Ці мережі повинні забезпечувати необхідні значення якості передачі інформації, і перш за все по затримці, пропускну здатності та безпеки. Цей рівень складається з конвергентної мережевої інфраструктури, яка створюється шляхом інтеграції різноманітних мереж в єдину мережеву платформу. Конвергентний абстрактний мережевий рівень у IoT дозволяє через відповідні шлюзи декільком користувачам використовувати ресурси в одній мережі незалежно і спільно без шкоди для конфіденційності, безпеки і продуктивності.

3. Сервісний рівень

Сервісний рівень містить набір інформаційних послуг, спрямованих автоматизувати технологічні і бізнес-операції в IoT: підтримки операційної та бізнес діяльності (OSS/BSS, Operation Support System/Business Support System), різної аналітичної обробки інформації (статистичної, інтелектуального аналізу даних і текстів, прогностична аналітика та ін), зберігання даних, забезпечення інформаційної безпеки, управління бізнес-правилами (BRM, Business Rule Management), управління бізнес-процесами (BPM, Business Process Management) і ін.

4. Рівень додатків

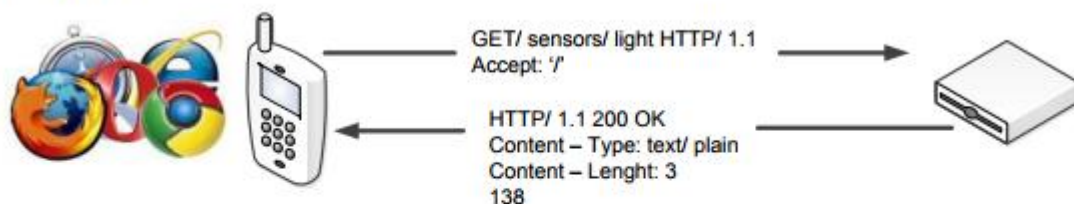
На четвертому рівні архітектури IoT існують різні типи додатків для відповідних промислових секторів і сфер діяльності (енергетика, транспорт, торгівля, медицина, освіта та ін.). Додатки можуть бути «вертикальними», коли вони є специфічними для конкретної галузі промисловості, а також «горизонтальними», (наприклад, управління автопарком, відстеження активів та ін), які можуть використовуватися в різних секторах економіки.

Складовою частиною Інтернету речей є Веб речей (WEB of Things, WoT), який забезпечує взаємодію різних інтелектуальних об'єктів («речей») з використанням стандартів і механізмів Інтернет, таких як уніфікований (стандартизований) ідентифікатор ресурсу URI (Uniform Resource Identifier), протокол передачі гіпертексту HTTP (HyperText Transfer Protocol), спосіб побудови архітектури розподіленого додатку REST (Representational State Transfer) та ін. Фактично WoT передбачає реалізацію концепції IoT на прикладному рівні з використанням вже існуючих архітектурних рішень, орієнтованих на розробку web-додатків.

Іншими словами, дані з розумних речей або управління ними повинно бути доступно через WWW-сторінки. На рис. 2 показано, як, використовуючи спеціальну сторінку в Інтернет через браузер можна зчитати дані з датчика світла в бездротовій сенсорній мережі або змінити колір четвертого індикатора в сенсорі.

Читання інформації з сенсора, наприклад зчитування показань датчика світла WEB-Клієнт.

Читання інформації з сенсора, наприклад зчитування показань датчику світла
WEB-клієнт



Управління акаунтом, наприклад, зміна кольору світлодіоду
WEB-клієнт

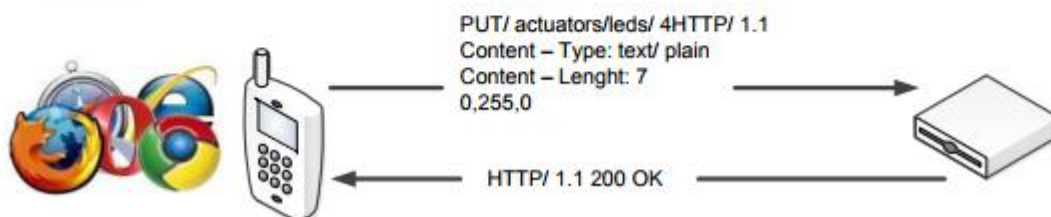


Рис. 2 Приклади веб-взаємодії з пристроями сенсорної мережі

Основні властивості WoT:

1. Використовує протокол HTTP для доповнення, а не в якості транспортного механізму передачі даних, як він застосовується для традиційних WWW-послуг.
2. Забезпечує синхронну роботу інтелектуальних (смарт) об'єктів через прикладний програмний інтерфейс REST (також відомий як RESTful API) і в цілому відповідає ресурсно-орієнтованій архітектурі ROA (Resource-Oriented Architecture).
3. Надає асинхронний режим роботи інтелектуальних об'єктів з використанням значною мірою стандартних Web-технологій, таких як Atom, містить формат для опису ресурсів веб-сайтів і протокол для їх публікації або Web-механізмів передачі даних, таких як модель роботи веб-додатку Comet, при якому постійне HTTP-з'єднання дозволяє веб-серверу відправляти дані браузеру без додаткового запиту з боку браузера.

Ці характеристики WoT забезпечують просту взаємодію інтелектуальних об'єктів через Інтернет, крім того вони реалізують однаковий інтерфейс для доступу і підтримки функціональності смарт-об'єктів.

З концепцією WoT перегукується ідея Семантичної павутини (Semantic Web) – це напрямок розвитку Всесвітньої павутини WWW, метою якого є представлення інформації у вигляді, придатному для машинної обробки. Термін «семантична павутина» був вперше відокремлений Тімом Бернерсом-Лі (винахідником Всесвітньої павутини) у травні 2001 року. Концепція семантичної павутини була прийнята і впроваджується Консорціумом Всесвітньої павутини W3C (World Wide Web Consortium).

У звичайній Павутині, заснованій на HTML-сторінках, інформація закладена в тексті сторінок і витягується людиною за допомогою браузера. Семантична ж павутина передбачає запис інформації у вигляді семантичної мережі за допомогою онтологій. Під онтологією розуміється формальний явний опис понять у розглянутій предметній області (класів). Онтологія разом з набором індивідуальних екземплярів класів утворює базу знань. Таким чином, програма-клієнт може безпосередньо отримувати з павутини факти і робити з них логічні висновки. Семантична павутина працює паралельно зі звичайною Павутиною і на її основі, використовуючи протокол HTTP і ідентифікатори URI.

Незважаючи на всі переваги, що надаються семантичною павутиною в разі її впровадження, існують певні сумніви в можливості її повної реалізації.

Вказуються різні причини, які можуть бути перешкодою до цього, починаючи з людського фактора (люди схильні уникати роботи по підтримці документів з метаданими, відкритими залишаються проблеми істинності метаданих тощо). Крім того необхідність опису метаданих так чи інакше призводить до дублювання інформації.

Кожен документ повинен бути створений в двох примірниках: розмічених для читання людьми, а також у машинно-орієнтованому форматі.

Використовують 3 способи взаємодії з інтернет-речами:

- 1) прямий доступ;
- 2) доступ через шлюз;
- 3) доступ через сервер.

У разі прямого доступу інтернет-речі повинні мати власну IP-адресу або мережевий псевдонім, за яким до них можна звернутися з будь-якого клієнтського додатку і вони повинні виконувати функції веб-сервера. Інтерфейс з такими речами зазвичай виконаний у вигляді web-ресурсу з графічним інтерфейсом для управління за допомогою веб-браузера. Можливе використання спеціалізованого програмного забезпечення. В такі веб-пристрої повинен бути інтегрований прикладний програмний інтерфейс RESTful API для прямого доступу до них через Інтернет.

Недоліки такого способу очевидні: необхідність мати фіксовану адресу в мережі, що залежить від провайдера послуги зв'язку з Інтернетом таких речей; іншим виходом з ситуації є використання мережевого псевдоніма IP-адреси (alias), що вимагає постійного звернення інтернет-речі до спеціального сервера із запитом про оновлення мережевої адреси за псевдонімом; ліміт підключень до пристрою – викликано низькою якістю зв'язку інтернет-речей, а також їх слабкими обчислювальними ресурсами. Така проблема вирішується шляхом включення до складу інтернет-речі високопродуктивного обладнання і підключення речей до стабільного джерела зв'язку з Інтернетом. Це викликає необхідність у більшому споживанні енергії такою річчю і часто змушує робити такі речі стаціонарними, що живляться від постійних джерел електроенергії.

Якщо інтернет-речі не мають вбудованої підтримки протоколів IP і HTTP, а підтримують приватні протоколи, наприклад Bluetooth або ZigBee, то для взаємодії з ними можна використовувати спеціальний Інтернет-шлюз. Він є веб-сервером, який через інтерфейс REST API взаємодіє з IP-пристроями, і перетворює запити, що надходять від них до специфічного API пристрою, підключеного до цього шлюзу. Основна перевага використання Інтернет шлюзу в тому, що він може підтримувати кілька типів пристроїв, що використовують власні протоколи для зв'язку.

Доступ до інтернет-речей через шлюз є більш раціональним способом організації взаємодії і повністю витісняє метод прямого доступу в разі необхідності організації зв'язку бездротових сенсорних мереж або мережі Інтернет-речей з глобальною мережею Інтернет. Більшість стандартів бездротових сенсорних мереж не підтримують протокол IP, використовуючи власні протоколи взаємодії. Така особливість викликає необхідність наявності пристрою для ретрансляції повідомлень з сенсорної мережі в мережу Інтернет для сумісності протоколів.

Недоліки такого підходу ті ж, що й у випадку прямого доступу, але поширюються вони вже на шлюз.

Третя форма взаємодії пристроїв у IoT через сервер передбачає наявність посередника між інтернет-речами і користувачем і може бути реалізована за допомогою посередницької платформи даних. Даний підхід передбачає наявність централізованого сервера або групи серверів, в основні функції яких входить: приймання повідомлень від інтернет-речей і передача їх користувачам; зберігання прийнятої інформації та її обробка; забезпечення користувацького інтерфейсу з можливістю двостороннього обміну між користувачем та інтернет-річчю.

Основною метою використання посередницьких платформ даних є спрощення пошуку, контролю, візуалізації та обміну даними з різними речами». В основі даного підходу лежить централізоване сховище даних. Кожен пристрій, що має доступ в мережі Інтернет (прямий або через інтернет-шлюз), повинен бути зареєстрованим в системі, перш ніж він зможе почати передачу даних. При цьому істотно знижуються вимоги до продуктивності пристроїв, так як від них не вимагається виконання функцій веб-сервера. Набір інструментів, що надаються платформами, істотно спрощує розробку нових додатків для взаємодії та управління об'єктами WoT.

Такий спосіб доступу є найбільш раціональним і часто використовуваним, оскільки дозволяє перенести навантаження обробки запитів користувачів з інтернет-речей на централізований сервер, тим самим розвантажуючи слабкий радіоканал зв'язку інтернет-речей, переносючи навантаження на дротові канали зв'язку між сервером і користувачами.

Метод централізованого сервера також надає надійні засоби зберігання і обробки інформації, що дозволяє інтернету-речей взаємодіяти один з одним і користуватися хмарними обчисленнями. Даний підхід може використовувати також метод шлюзу для з'єднання локальних бездротових мереж з сервером.

В Інтернеті речей шлюз використовується не тільки для прямого зв'язку інтернет-речей з користувачем, але і при використанні централізованого сервера. Шлюзи служать засобом для об'єднання локальних мереж інтернет-речей з глобальною мережею і зв'язком з сервером управління або кінцевим користувачем. Оскільки локальні мережі інтернет-речей являють собою в основному бездротові сенсорні мережі, то шлюзи, використовувані в Інтернеті речей, аналогічні використовуваним в територіально-розподілених сенсорних мережах. Існує кілька способів організації шлюзів.

Перший спосіб полягає у використанні комп'ютерів, які мають точку доступу до глобальної мережі Інтернет, і кожна з об'єднаних мереж підключена до такого комп'ютера. Основними недоліками такого підходу є вартість і громіздкість.

Сенсорні мережі складаються з мініатюрних датчиків і повинні працювати автономно, однак територіально-розподілена сенсорна мережа при такому підході втрачає властивість автономності, оскільки тепер вона залежить від наявності електрики і точки доступу Інтернет на комп'ютері.

Другий спосіб полягає у використанні пристрою-шлюзу, що дозволяє з'єднати сенсорну мережу з найближчою провідною мережею, що має вихід в Інтернет. Такою провідною мережею, як правило, є Ethernet-мережа.

Третій спосіб полягає у використанні пристрою-шлюзу, який є повністю автономним і саме надає точку доступу до мережі Інтернет. Це можливо при використанні бездротових технологій передачі даних. Пристрій складається з одного передатчика, сумісного з сенсорною мережею і другого – сумісного з тією або іншою глобальною бездротовою мережею, сфера дії якої потрапляє сенсорна мережа. Такими мережами можуть служити GSM або WiMAX. Використання мережі GSM є більш економічним в плані енергоспоживання.

Існують також шлюзи, що надають доступ сенсорним мережам до найближчих мереж Wi-Fi для пошуку точки доступу до мережі Інтернет.

Таким чином, якщо необхідно організувати повністю автономну територіально-розподілену сенсорну мережу, то слід використовувати третій спосіб.

Якщо ж сенсорна мережа використовується як частина якоїсь великої провідної мережі, то немає необхідності в її повній автономності і можливо використання перших двох способів.

Висновки. Платформи IoT – це найвищий рівень інтелекту і призначеного для користувача інтерфейсу, який пов'язує підключені пристрої і веб-сервіси. Вони спільно визначають модель еталонної архітектури для IoT, беручи до уваги широкий спектр технологій, комунікаційних протоколів і стандартів. Платформа IoT повинна дозволяти зовнішнім користувачам або пристроям підключатися до неї, ґрунтуючись на моделі управління, яка є основою для прийняття рішення. Платформи IoT здатні координувати і керувати проблемами з підключенням, а також гарантувати безпеку і конфіденційність даних, якими обмінюється велика кількість мережевих пристроїв, одночасно долаючи проблеми взаємодії. Використання визначеного набору протоколів для обміну певними службами, об'єднання платформ дозволить оптимізувати використання ресурсів, поліпшити якість обслуговування і, швидше за все, скоротити витрати. Платформи IoT повинні мінімізувати складність збору і обробки великих обсягів даних, що генеруються в сценаріях IoT.

Пропонуючи інноваційні рішення, які забезпечують можливість самоврядування, самоконтролю і масштабованості платформи IoT дозволяють підприємствам створювати додатки для задоволення потреб цифрового бізнесу і стануть важливим елементом у розвитку цифрового єдиного ринку.

Підхід до розвитку інфраструктури «Інтернету речей» буде поетапним, що включає в себе розширення існуючих методів ідентифікації, таких як RFID.

1. «Інтернет вещей» в промышленности: обзор ключевых технологий и трендов // Ли Да Сюй (Li Da Xu), Ву Хе (Wu He) - whe@odu.edu, Шянчан Ли (Shancang Li) - shanchang.li@bristol.ac.uk, перевод Алексей Осотов. – <http://www.controlengrussia.com/internet-veshhej/klyuchevy-h-tehnologii/> (Станом на 10.04.2018)
2. Семенченко П.И. Обзор и анализ функциональных возможностей платформ для устройств интернета вещей / П.И. Семенченко // Российский экономический университет им. Г.В. Плеханова, 2017. – № 5. – С. 156-168.

3. Шварц Марко Интернет верей с ESP8266: Пер. с англ. – СПб.: БХВ-Петербург, 2018. – 192 с.
4. Li S., Xu L., Wang X. Compressed sensing signal and data acquisition in wireless sensor networks and internet of things // IEEE Trans. Ind. Informat. 2013. – Vol. 9. – No. 4.
5. He W., Xu L. Integration of distributed enterprise applications: A survey // IEEE Trans. Ind. Informat. 2014. – Vol. 10. – No. 1.
6. Uckelmann D., Harrison M., Michahelles F. An architectural approach towards the future internet of things // Uckelmann D., Harrison M., Michahelles F. Architecting the Internet of Things. USA, NY: Springer, 2011.
7. Wang L., Xu L., Bi Z., Xu Y. Data filtering for RFID and WSN integration // IEEE Trans. Ind. Informat. 2014. – Vol. 10. – No. 1.

УДК 004.056.55

Гринюк С.В., Поліщук М.М., Міскевич О.І., Харковець Р.В.
Луцький національний технічний університет**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ШИФРУВАННЯ ТА ДЕШИФРУВАННЯ ІНФОРМАЦІЇ
КРИПТОГРАФІЧНИМИ МЕТОДАМИ ЗАСОБАМИ VISUAL STUDIO**

Гринюк С.В., Поліщук М.М., Міскевич О.І., Харковець Р.В. Програмне забезпечення для шифрування та дешифрування інформації криптографічними методами засобами Visual Studio. У статті розглядаються проблеми захисту інформації в комп'ютерних системах та мережах. Дається класифікація алгоритмів шифрування та дешифрування інформації. На основі розглянутих алгоритмів шифрування було створено програмне забезпечення для шифрування та дешифрування файлів, використовуючи засоби Visual Studio.

Ключові слова: програмне забезпечення, криптографія, алгоритм, інформація.

Гринюк С.В., Поліщук М.М., Міскевич О.І., Харковець Р.В. Програмное обеспечение для шифрования и дешифрования информации криптографическими методами средствами Visual Studio. В статье рассматриваются проблемы защиты информации в компьютерных системах и сетях. Дается классификация алгоритмов шифрования и дешифрования информации. На основе рассмотренных алгоритмов шифрования было создано программное обеспечение для шифрования и дешифрования файлов, используя средства Visual Studio.

Ключевые слова: программное обеспечение, криптография, алгоритм, информация.

Hrynyuk S.V., Polishchuk M.M., Miskevich O.I., Kharkovets R.V. Software for encrypting and decrypting information by cryptographic methods with Visual Studio tools. This article deals with problems of information security in computer systems and networks. The classification of algorithms of encryption and decryption of information is given. Based on the encryption algorithms under consideration, software for encrypting and decrypting files was created using Visual Studio tools.

Key words: software, cryptography, algorithm, information.

Вступ. Однозначно та безперечно є той факт, що рівень розвитку будь якої держави та суспільства визначається рівнем їх інформатизації. У зв'язку з цим в Україні та й інших країнах світу новітні технології досить часто використовуються у всіх сферах діяльності людини. На даний момент не можливо представити своє життя без технологій. Сфера використання технологій досить широка. Всі підприємства, школи, банки, аеропорти, університети, атомні електростанції, різні міністерства використовують комп'ютери для працездатності, передаючи різного роду інформацію.

Постановка наукової проблеми. Основною проблемою є те, що необхідно захистити інформацію, яка передається між комп'ютерами. Досить недавно на прикладі вірусу Petya, всі люди переконались, що потрібно захищати свої системи, щоб ніхто в них не проникав. Також недавно виник такий термін як кібервійна. Для того щоб перемогти в ній, достатньо показати свою перевагу в інформаційній грамотності, в тому що ти вмєш захищати свої системи й знаєш як обійти системи захисту твого супротивника.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Вирішенням цієї проблеми є криптографія та алгоритми шифрування/дешифрування інформації. На основі алгоритмів шифрування я розробив програму за допомогою засобів Visual Studio на мові C# для шифрування та дешифрування файлів. Вона нам допоможе безпечно обмінюватись файлами між комп'ютерами. Використовувати ми будемо алгоритм шифрування AES.

Алгоритм шифрування AES використовується в досить популярному месенджері Telegram, в якому шифруються повідомлення проходячи кілька етапів. Враховуючи те, що Telegram був створений для обміну захищеними повідомленнями які важко перехватити, я вирішив створити прикладну програму на основі алгоритму AES, для шифрування файлів.

Advanced Encryption Standard (AES), також його часто називають Rijndael — симетричний алгоритм блочного шифрування (розмір блока 128 біт, ключ 128/192/256 біт). Прийнятий стандартом шифрування в США. AES має фіксовану довжину у 128 біт, а розмір ключа може приймати значення 128, 192 або 256 біт. Через фіксований розмір блоку AES оперує із масивом 4×4 байт. Для ключа 128 біт алгоритм має 10 раундів у яких послідовно виконуються операції:

SubBytes() – обробляє кожен байт стану, проводячи нелінійну заміну байтів використовуючи таблицю замінів S-box. Це забезпечує нелінійність шифрування;

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Рис.1 - S-Box таблиця замінів

Наприклад на вході маємо 54 то на виході будемо мати 20. Ця заміна буде повторюватись 10 разів. Після проходження всіх етапів які зазначені нижче.

ShiftRows(): це процедура перестановки байтів в рядках. Перший рядок залишається без змін, в другому рядку йде зміщення на байт, в наступному на два байта, а в останньому на три, як показано на наступному рис2:

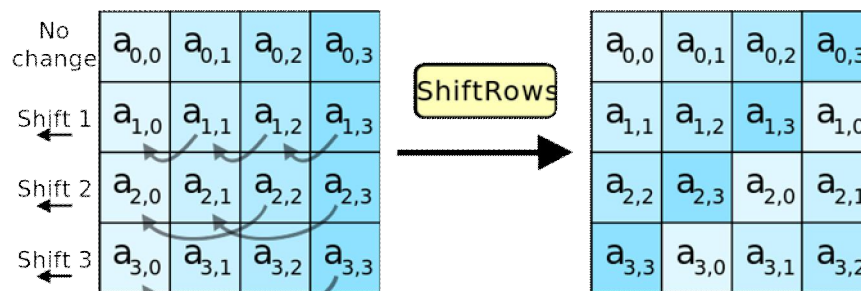


Рис. 2 – Процедура ShiftRows побітова заміна в рядках

У процедурі MixColumns, чотири байти кожної колонки статичних даних змішуються, використовуючи для цього зворотну лінійну трансформацію. MixColumns працює по колонках, трактуючи кожну з них як рівняння четвертого степеня. Над цими рівняннями виконується множення в $GF(2^8)$ по модулю $x^4 + 1$.

на фіксований многочлен $C(x) = 3x^2 + x^2 + x + 2$ та разом з ShiftRows MixColumns вносить це що вийде в шифр.

Під час цієї операції, кожен стовпчик множиться на матрицю, яка для 128-бітного ключа має вигляд:

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

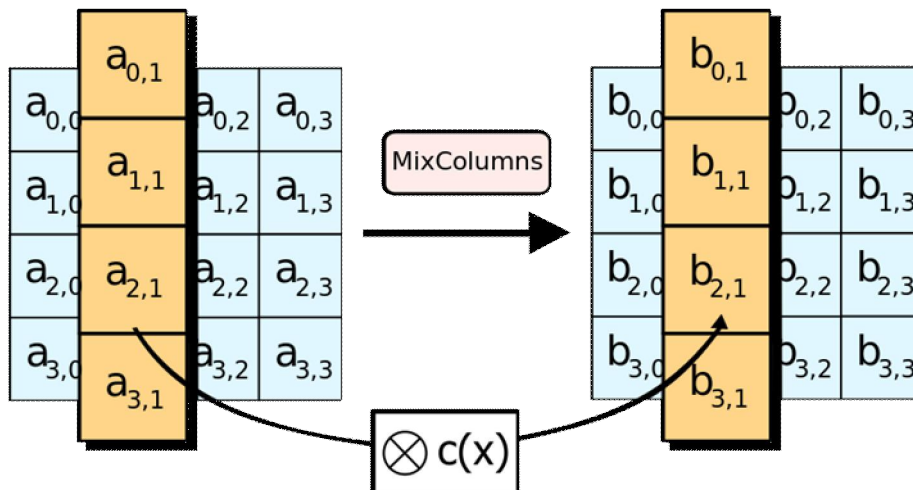


Рис. 3 - Приклад роботи процедури MixColumns

У процедурі AddRoundKey, RoundKey кожного раунду об'єднується зі статичними даними. Для кожного раунду RoundKey виходить з шифру ключа використовуючи процедуру KeyExpansion; кожен RoundKey такого ж розміру, що і статичні дані. Фактично це звичайний побайтовий XOR байт ключа з байтами таблиці статичних даних як показано на рисунку 3.

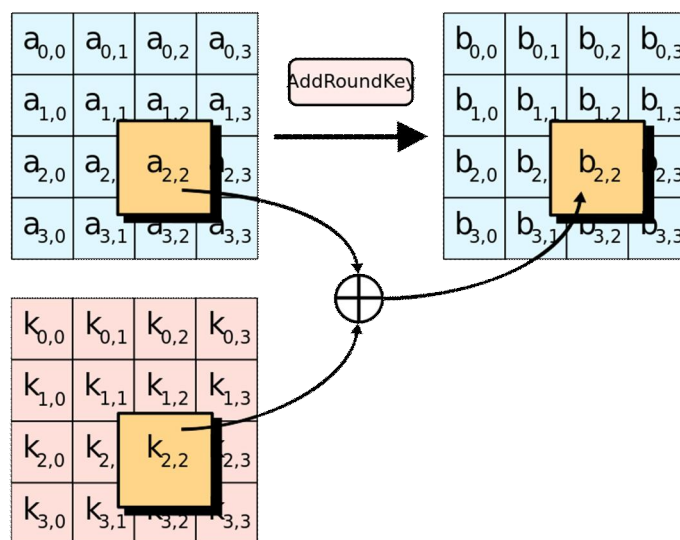


Рис. 4 - Процедура AddRoundKey

Робота алгоритму шифрування закінчиться тоді, коли пройде 10 разів всі ці процедури. В кінці ми отримуємо зашифрований файл, який без ключа не можливо буде розшифрувати. Знаючи ключ ми зможемо запустити алгоритм розшифрування.

Спираючись на цей алгоритм була розроблена програма CryptoCode 1.0 (Рис.5).

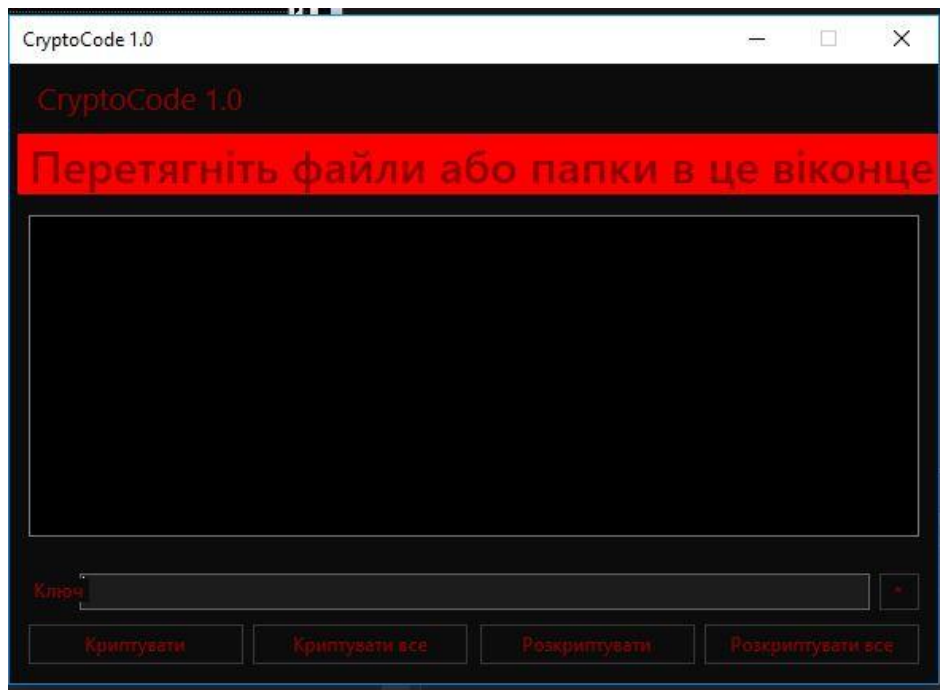


Рис. 5 - Вікно програми CryptoCode 1.0

Щоб реалізувати AES програмно потрібно підключити простір імен для криптографії:

```
using System.Security.Cryptography;
```

Приклад AES шифрування реалізували так :

```
public static byte[] Encrypt(byte[] rawBytes, string password)
{
    RijndaelManaged AES = new RijndaelManaged();
    byte[] hash = new byte[32];
    byte[] temp = new
MD5CryptoServiceProvider().ComputeHash(Encoding.ASCII.GetBytes(password));
    System.Buffer.BlockCopy(temp, 0, hash, 0, 16);
    System.Buffer.BlockCopy(temp, 0, hash, 15, 16);
    AES.Key = hash;
    AES.Mode = CipherMode.ECB;
    ICryptoTransform AESEncryptor = AES.CreateEncryptor();
    return AESEncryptor.TransformFinalBlock(rawBytes, 0, rawBytes.Length);
}
```

Розшифрування всього алгоритму виконано так :

```
public static byte[] Decrypt(byte[] rawBytes, string password)
{
    RijndaelManaged AES = new RijndaelManaged();
    byte[] hash = new byte[32];
    byte[] temp = new
MD5CryptoServiceProvider().ComputeHash(Encoding.ASCII.GetBytes(password));
```

```
System.Buffer.BlockCopy(temp, 0, hash, 0, 16);  
System.Buffer.BlockCopy(temp, 0, hash, 15, 16);  
AES.Key = hash;  
AES.Mode = CipherMode.ECB;  
ICryptoTransform AESDecrypter = AES.CreateDecryptor();  
return AESDecrypter.TransformFinalBlock(rawBytes, 0, rawBytes.Length);  
}
```

Перетягуємо файл який хочемо зашифрувати у вікно програми, вводимо ключ (хоча це не обов'язково) й натискаємо кнопку «Криптувати» й отримуємо зашифрований файл:

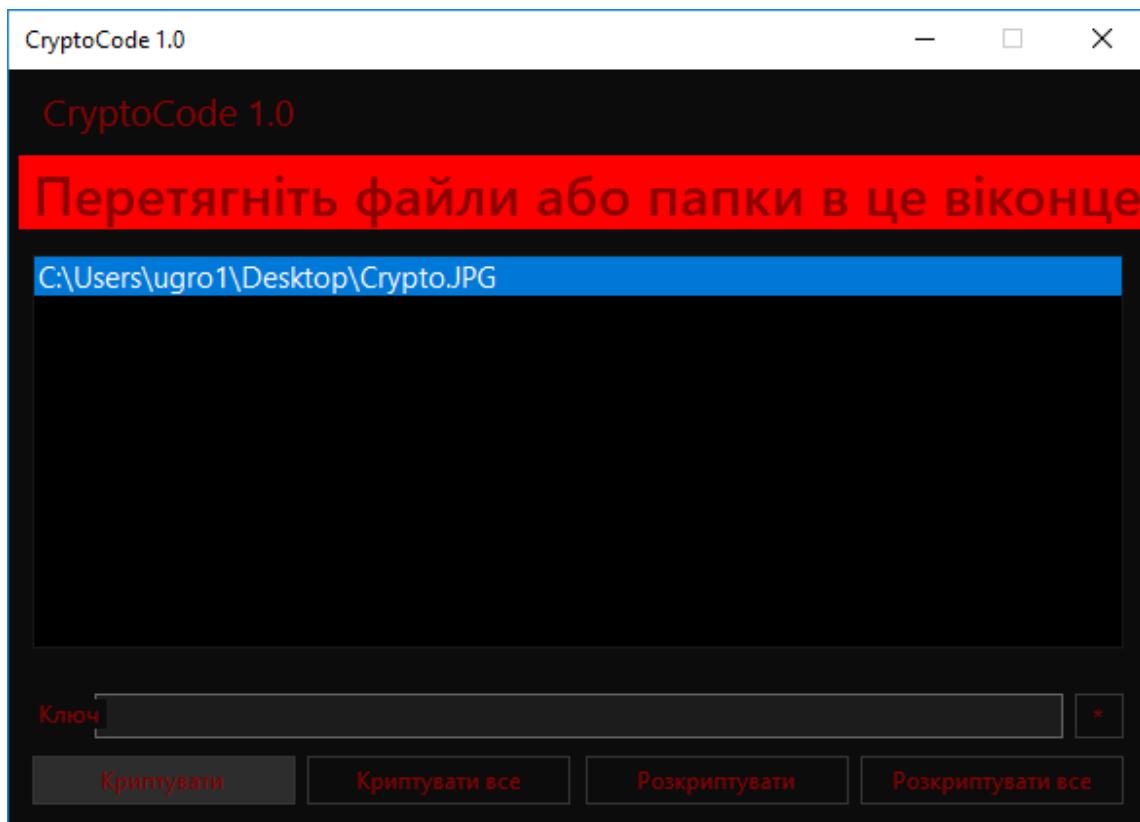


Рис. 6 - Вікно програми та зашифрований файл

Перетягуємо зашифрований файл назад у вікно програми вводимо в поле «Ключ» правильний ключ, який ми вказували перед шифруванням, і нажимаємо «Розкриптувати». Як можемо бачити на Рис.7, що розшифрувавши, ми отримали файл з розширенням .JPG .

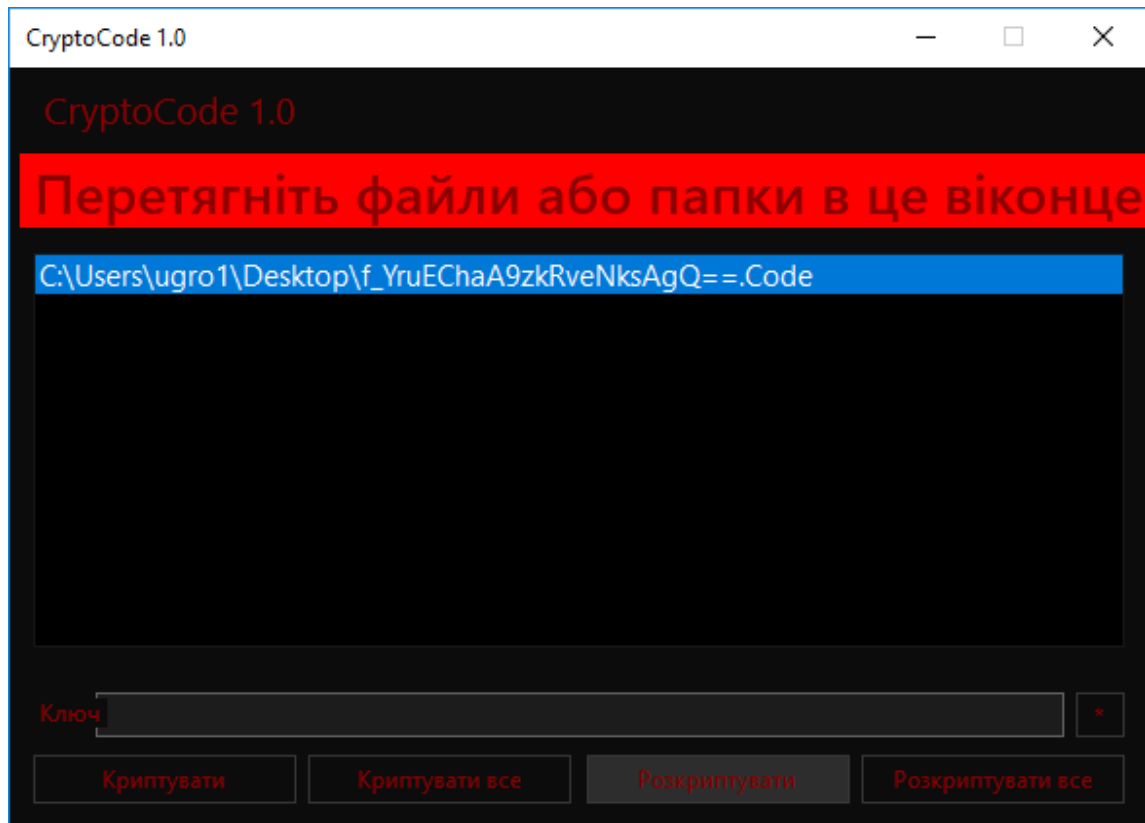


Рис. 7 - Розшифрування зашифрованого файла

Також основною особливістю програми є те, що при введенні невірної ключа зашифрований файл видаляється з комп'ютера. Можна також підмітити те, що зашифрований файл можна ще пару раз зашифровувати змінюючи ключ.

Висновок. У підсумку можна сказати те, що необхідно захищати файли, системи, мережі від несанкціонованого доступу. Алгоритм шифрування AES, який є стандартом в США дозволяє захистити необхідну інформацію. Захист інформації є перспективним напрямком, оскільки це забезпечує тебе впевненістю в тому, що твоє приватне не стане публічним.

1. Баричев С. Г., Гончаров В. В., Серов Р. Е. 2.4.2. Стандарт AES. Алгоритм Rijdael // Основы современной криптографии — 3-е изд. — М.: Диалог-МИФИ, 2011. — С. 30–35. — 176 с.
2. Mollin, R. A. Introduction to Cryptography. — CRC Press, 2007. — P. 80. — 413 p.
3. [https://msdn.microsoft.com/ru-ru/library/system.security.cryptography.aes\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.security.cryptography.aes(v=vs.110).aspx)
4. https://uk.wikipedia.org/wiki/Advanced_Encryption_Standard

УДК 004.434

Г. М. Губаль

Луцький національний технічний університет

ВИКОРИСТАННЯ ДЕЯКИХ КОМАНД МОВИ L^AT_EX ДЛЯ СТВОРЕННЯ МАТЕМАТИЧНИХ ТЕКСТІВ

Губаль Г.М. Використання деяких команд мови L^AT_EX для створення математичних текстів. У статті наведено і проаналізовано деякі питання створення математичних формул, математичних тестів, водяних знаків у документі L^AT_EX, створення посилань і гіперпосилань засобами L^AT_EX. Досліджено деякі можливості стильового пакету setspace, які є актуальними при створенні математичних текстів.

Ключові слова: система L^AT_EX, numprint, dashundergaps, водяний знак, \vphantom.

Літ. 15.

Губаль Г.Н. Использование некоторых команд языка L^AT_EX для создания математических текстов. В статье приведено и проанализировано некоторые вопросы создания математических формул, математических тестов, водяных знаков в документе L^AT_EX, создание ссылок и гиперссылок средствами L^AT_EX. Исследовано некоторые возможности стилевого пакета setspace, актуальные при создании математических текстов.

Ключевые слова: система L^AT_EX, numprint, dashundergaps, водяной знак, \vphantom.

Лит. 15.

Hubal H.M. Using some commands of the L^AT_EX language for the creation of mathematical texts. Some questions for creation of mathematical formulae, of mathematical tests, of watermarks in the L^AT_EX document, creation of references and hyperreferences by means of L^AT_EX are given and analyzed in the paper. Some opportunities of the setspace style package are investigated, relevant when generating mathematical texts.

Keywords: system L^AT_EX, numprint, dashundergaps, watermark, \vphantom.

Bibl. 15.

Вступ. Робота з системою L^AT_EX, призначеною для створення математичних текстів [1, 2, 5-10, 12-15] подібна до програмування, тим більше, що ця система розрахована на використання текстового режиму, командного рядка і багатьох конфігураційних файлів [3, 4, 11].

У цій статті наведемо і проаналізуємо деякі питання створення математичних формул, математичних тестів, водяних знаків у документі L^AT_EX, створення посилань і гіперпосилань засобами L^AT_EX. Досліджено деякі можливості стильового пакету setspace, які є актуальними при створенні математичних текстів.

Основна частина. У математичному режимі L^AT_EX документів числа пишуться, за замовчуванням, прямим шрифтом, а не курсивом. У текстовому режимі необхідно при написанні чисел як арабських, так і римських переходити до математичного режиму, оскільки, якщо речення записане курсивом, то математичний режим згенерує прямий шрифт числа. Наприклад, запис:

Тіло переміщається \emph{щонайбільше на \$2\$ км} на північ.

генерує такий текст:

Тіло переміщається *щонайбільше на 2 км* на північ.

У англійській літературі кожні три цифри розділяють комами:

240,121

1.542, 119

–2.128,5

У наукових документах кому замінюють тонким пробілом \, що може бути зроблено автоматично командою \np пакета numprint, наприклад, команда

\np{230000000}

генерує запис:

230,000,000

Зауважимо, що цей розділовий знак може бути крапкою, тонким пробілом або тільдою залежно від мови, заданої пакетом babel.

Пакет numprint генерує десятковий символ, що відповідає мові, заданій пакетом babel, а саме, крапку, якщо задана англійська мова і кому для іншої європейської мови. Наприклад, для задання англійської мови команда

`\np{7,29}`

генерує запис:

7.29

Цей пакет дозволяє заокруглити число до заданої кількості цифр. Наприклад, якщо необхідно заокруглити число 3,654711205 до трьох цифр, то команда

`\np{3,654711205}`

генерує запис:

3.655

Крім того, стильовий пакет `numprint` у разі необхідності дописує нулі. Наприклад, команда

`\np{,271}`

генерує запис:

0.271

При друці інтегралів не завжди висота знака інтеграла адаптована до висоти підінтегральної функції (виразу). Наприклад, формула

$$\int \frac{dx}{\sqrt{\left(x + \frac{1}{2}\right)^2 + \frac{3}{4}}}$$

генерується за допомогою такого коду:

`\begin{equation*}`

`\int\frac{dx}{\sqrt{\left(x+\frac{1}{2}\right)^2+\frac{3}{4}}}`

`\end{equation*}`

Для адаптації висоти знака інтеграла до висоти підінтегральної функції (виразу) можна використовувати команду `\displaystyle`. Наприклад, код

`\begin{equation*}`

`\displaystyle\int\frac{dx}{\sqrt{\left(x+\frac{1}{2}\right)^2+\frac{3}{4}}}`

`\end{equation*}`

генерує таку формулу:

$$\int \frac{dx}{\sqrt{\left(x + \frac{1}{2}\right)^2 + \frac{3}{4}}}$$

Зауважимо, щоб знак інтеграла був пов'язаний з підінтегральною функцією (виразом), можна використати команди стильового пакету `bigints`, попередньо підключивши його у преамбулі документа. При використанні команд `\bigint` (для знаку інтеграла $\int$) і `\bigoint` (для знаку інтеграла по замкнутому контуру $\oint$) використовується такий принцип: чим більше букв "s" дописувати до команди інтеграла (`\bigints`, `\bigintss`, `\bigoints`, `\bigointss` і т.д.), тим меншу висоту набуває знак інтеграла.

Для створення математичних тестів, в яких необхідно підкреслити текст або виділити місце, в яке студент повинен вписати правильну відповідь можна використовувати стильовий пакет `dashundergaps`. Цей пакет використовує макроси для підкреслення тексту точками, пунктиром і суцільною лінією. При цьому необхідно приєднати даний пакет у преамбулі документа.

Деякі опції стильового пакета `dashundergaps` використовуються з відповідними командами:

- 1) Опція `dot`, використана з командою `\dotuline{text}`, підкреслює `text` точками: text;
- 2) Опція `dash`, використана з командою `\dashuline{text}`, підкреслює `text` пунктиром: text;
- 3) Опція `phantomtext`, використана з командою `\gap{text}`, генерує на місці тексту:
 - шаблон, підкреслений суцільною лінією, якщо використовуються опції `dash` і `dot` або не використовується жодна з них: _____ (1);
 - шаблон, підкреслений точками, якщо використовується опція `dot`:(1);
 - шаблон, підкреслений пунктиром, якщо використовується опція `dash`: _ _ _ _ (1);
 - сам `text`, якщо використовується опція `teachernotes`: `text`.

Щоб змінити міжрядковий інтервал для фрагменту тексту (на одинарний, розміром у 1,5 рядка, подвійний), необхідно у преамбулі документа підключити стильовий пакет `setspace`:

`\usepackage{setspace}`

% `\singlespacing` % одинарний інтервал для тексту всього документа

% або `\onehalfspacing` % інтервал розміром у 1,5 рядка для тексту всього документа

% або `\doublespacing` % подвійний інтервал для тексту всього документа

% або `\setstretch{множник}` % довільний інтервал для тексту всього документа

Тоді у тілі документа створюємо код:

`\begin{onehalfspace}`

Фрагмент тексту з інтервалом розміром у 1,5 рядка

`\end{onehalfspace}`

`\begin{doublespace}`

Фрагмент тексту з подвійним міжрядковим інтервалом

`\end{doublespace}`

`\begin{spacing}{0.9}`

Фрагмент тексту з міжрядковим інтервалом 0,9

`\end{spacing}`

Команди `\singlespacing`, `\onehalfspacing`, `\doublespacing` і `\setstretch{множник}` у преамбулі задають міжрядковий інтервал тексту всього документа.

Для документів, що друкуються на обох сторонах аркуша (для двостороннього друку), а також для інших типів документів використовується стильовий пакет `turnthepage`. Для цього необхідно приєднати цей пакет у преамбулу документа командою:

`\usepackage[<option>]{turnthepage}`

Option генерує текст, який з'являється у правому нижньому куті непарної сторінки.

Для стильового пакета `english` генерується такий текст:

1) `'/...'`

2) `'Turn the page'`

Якщо необхідно змінити текст у правому нижньому куті сторінки, можна перевизначити команду `\turnthepage` після преамбули документа. Наприклад,

`\renewcommand{\turnthepage}{%`

`\tshape` (Будь-ласка, перегорніть сторінку)

Розглянемо питання створення водяних знаків у документі $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, які використовуються або з міркувань безпеки, або вказати важливу інформацію стосовно документа.

У $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ документ можна вбудувати водяні знаки, використовуючи такі стильові пакети:

1) `xwatermark`;

2) `draftwatermark`;

3) `draftcopy`;

4) `TikZ`.

Використовуючи стильовий пакет `xwatermark` для створення водяного знаку у $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ документі, необхідно у преамбулі цього документа записати такий код:

`\usepackage[printwatermark=true, allpages=true, fontfamily=pag, color=gray, grayness=0.7, mark=watermark1, angle=90, fontsize=4 cm, markwidth=\paperwidth, fontseries=b, scale=0.8, xcoord=0, ycoord=0]{xwatermark}`

Використовуючи стильовий пакет `draftwatermark` для створення водяного знаку у $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ документі, необхідно у преамбулі документа створити такий код:

`\usepackage{draftwatermark}`

`\setwatermarklightness{0.4}`

`\setwatermarkangle{90}`

`\setwatermarkfontsize{4cm}`

`\setwatermarkscale{0.8}`

`\setwatermarktext{watermark1}`

Використовуючи стильовий пакет `draftcopy`, у преамбулі документа необхідно записати:

`\usepackage[english, all, portrait, draft]{draftcopy}`

Щоб використати стильовий пакет TikZ, необхідно його завантажити у преамбулі документа. За допомогою стильового пакету TikZ можна створити водяний знак watermark1 у центрі сторінки L^AT_EX документа:

```
\begin{tikzpicture}[remember picture, overlay]
\node[rotate=0, scale=14, text opacity=0.2] at (current page.center) {watermark1}
\end{tikzpicture}
```

Щоб створити водяний знак watermark1 у боксі сторінки L^AT_EX документа за допомогою стильового пакету TikZ, треба записати такий код:

```
\begin{tikzpicture}[remember picture, overlay]
\node[xdiff=1cm, yshift=1cm] at (current page.north east)
[textwidth=6cm, fill=blue:20, rounded corners above right]
{
watermark1
}
\end{tikzpicture}
```

Для полегшення роботи з pdf-документом, згенерованим з tex-документа, зручно створювати посилання і гіперпосилання у тексті. Для цього необхідно використовувати стильовий пакет `hyperref` для реалізації гіперпосилань у pdf-документі і стильовий пакет `xcolor` для виділення цих гіперпосилань кольором. Підключивши ці стильові пакети у преамбулі документа, створюємо такий код:

```
\usepackage{hyperref}
\usepackage{xcolor}
\definecolor{linkcolor}{HTML}{FF00FF}% колір посилань
\definecolor{urlcolor}{HTML}{FF00FF}% колір гіперпосилань
\hypersetup{pdfstartview=Fit, linkcolor=linkcolor, urlcolor=urlcolor, colorlinks=true}
```

Параметр `Fit` показує всю сторінку, центровану, розмір якої підігнаний під розмір вікна. Параметри `linkcolor` і `urlcolor` визначають кольори посилань і гіперпосилань, параметр `colorlinks=true` задає відображення посилань кольором, а не рамкою.

У тілі документа створюємо такий код:

$$\frac{d}{dx_2} \left(\mathbb{R}^{\nu} \times \left(\mathbb{R}^{\nu} \times \left(\sum_{i=1}^2 \frac{p_i^2}{2} + \Phi(q_1 - q_2) \right) \right) \right) = \frac{d}{dx_2} \left(\mathbb{R}^{\nu} \times \left(\mathbb{R}^{\nu} \times \left(\sum_{i=1}^2 \frac{p_i^2}{2} \right) \right) \right).$$

який генерує таку нумеровану формулу:

$$\int_{\mathbb{R}^V \times \mathbb{R}^V} dx_2 \{H_2, F_2(t, x_1, x_2 | F_1(t))\} = \int_{\mathbb{R}^V \times \mathbb{R}^V} dx_2 \left\{ \sum_{i=1}^2 \frac{p_i^2}{2} + \Phi(q_1 - q_2), \right. \\ \left. F_2(t, x_1, x_2 | F_1(t)) \right\}. \quad (1)$$

Посилання на номер формули здійснюється такою командою:

 $\text{\ref{formula1}},$

а посилання на формулу з написанням довільного тексту записується такою командою:

$$\backslash\mathrm{hyperref}[formula1]\{\text{перша формула}\}$$

Посилання на сайт здійснюється такою командою:

`\href{http://www.wikipedia.org}{wikipedia}`

Висновки. Таким чином, у цій статті наведено і проаналізовано деякі питання створення математичних формул, математичних тестів, вбудовування водяних знаків у документ L^AT_EX, створення посилань і гіперпосилань засобами L^AT_EX, а також досліджено деякі можливості стильового пакету setspace, які є актуальними при створенні математичних текстів.

1. Балдин Е. М. Компьютерная типография L^AT_EX / Е. М. Балдин. – СПб.: БХВ-Петербург, 2008.
2. Беляков Н. С. T_EX для всех / Н. С. Беляков, В. Е. Палощ, П. А. Садовский. – М.: Книжный дом «Либроком», 2009.
3. Губаль Г. М. L^AT_EX як видавнича система для створення математичних текстів і для програмування / Г. М. Губаль // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2013. – № 12.
4. Дубинич В. Н. Использование системы L^AT_EX для подготовки научных изданий / В. Н. Дубинич, М. В. Дубинич // Перспективы развития высшей школы: материалы V Международной науч.-метод. конф., Гродно: ГГАУ, 2012.
5. Жуков М. Ю. Оформление математических текстов при помощи пакета L^AT_EX2_ε / М. Ю. Жуков, Е. В. Ширяева. – Ростов н/Д: Изд-во ООО «ЦВВР», 2003.
6. Жуков М. Ю. L^AT_EX2_ε: искусство набора и вёрстки текстов с формулами / М. Ю. Жуков, Е. В. Ширяева. – Ростов н/Д: Изд-во ЮФУ, 2009.
7. Каров П. Шрифтовые технологии. Описание и инструментарий / П. Каров. Перевод с английского Карпинского О. С. и Куликова И. И. под редакцией, с предисловием и дополнением Ефимова В. В. – М.: Мир, 2001.
8. Кнут Д. Е. Всё про T_EX / Д. Е. Кнут. Пер. с англ. М. В. Лисиной. – Протвино: АО RDT_EX, 1993.
9. Котельников И. А. L^AT_EX по-русски / И. А. Котельников, П. З. Чеботаев. – Новосибирск: Сибирский хронограф, 2004.
10. Львовский С. М. Набор и вёрстка в системе L^AT_EX / С. М. Львовский. – 3-е изд., испр. и доп. – М.: МЦНМО, 2003.
11. Ширяева Е. В. Введение в T_EX–программирование / Е. В. Ширяева, И. В. Ширяева. – Ростов н/Д: Изд-во ЮФУ, 2010.
12. Goossens M. The L^AT_EX companion / M. Goossens, F. Mittelbach, A. Samarin. – Addison-Wesley, 1994. Русский перевод: Гуссенс М. Путеводитель по пакету L^AT_EX и его расширению L^AT_EX2_ε / М. Гуссенс, Ф. Миттельбах, А. Самарин. Перевод с английского Маховой О. А., Третьякова Н. В., Тюменцева Ю. В. и Чистякова В. В. под редакцией Маховой И. А. – М.: Мир, 1999.
13. Kopka H. Guide to L^AT_EX / H. Kopka, P. Daly. – Addison-Wesley, 2004.
14. Lamport L. L^AT_EX. A document preparation system, user's guide and reference manual / L. Lamport. – Addison-Wesley, 1994.
15. Spivak M. The joy of T_EX. A gourmet guide to typesetting with the A_MS-T_EX macro package. – American mathematical society, Providence, RI, 1990. Русский перевод: Спивак М. Восхитительный T_EX: руководство по комфортному изготовлению научных публикаций в пакете A_MS-T_EX / М. Спивак. – М.: Мир, 1993.

УДК 004.03:621.316.5

Журавська І.М.

Чорноморський національний університет імені Петра Могили

ІОТ-МЕРЕЖА НА БАЗІ BLUETOOTH-МОДУЛІВ ДЛЯ АВТОМАТИЗОВАНОГО КЕРУВАННЯ СПОЖИВАННЯМ ЕНЕРГОРЕСУРСІВ

Журавська І.М. ІоТ-мережа на базі Bluetooth-модулів для автоматизованого керування споживанням енергоресурсів. У статті запропонований алгоритм відключення від електроживлення неперіоритетних пристроїв за допомогою розроблених IoT Master та IoT Slave блоків. Керування блоками здійснюється або автоматизовано у локальній мережі, або віддалено через Інтернет. Запропонований підхід дозволяє запобігти надлишковому споживанню електроенергії та перенавантаженню мережі електроживлення при нормованому (або обмеженому за потужністю) споживанню енергоресурсів. Розроблена відповідна програмно-апаратна система на платформі Arduino з керуванням за протоколом Bluetooth.

Ключові слова: IoT блок, Arduino, Bluetooth, обмеження енергоспоживання, неперіоритетний прилад, автоматизоване відключення від електромережі.

Журавская И.Н. IoT-сеть на базе Bluetooth-модулей для автоматизированного управления потреблением энергоресурсов. В статье предложен алгоритм отключения от электропитания неперіоритетных устройств с помощью разработанных IoT Master и IoT Slave блоков. Управление блоками осуществляется либо автоматизировано в локальной сети, либо удаленно через Интернет. Предложенный подход позволяет предотвратить избыточное потребление электроэнергии и перегрузки сети электропитания при нормированном (или ограниченном по мощности) потреблении энергоресурсов. Разработана соответствующая программно-аппаратная система на платформе Arduino с управлением по протоколу Bluetooth.

Ключевые слова: IoT блок, Arduino, Bluetooth, ограничение энергопотребления, неперіоритетный прибор, автоматизированное отключение от электросети.

Zhuravska I.M. IoT network based on Bluetooth modules for automatized control power consumption. The article suggests the algorithm for unplugging non-priority devices from electric power consumption. The IoT Master and IoT Slave blocks have developed. Blocks' control is carried out either automatized in the local network, or remotely via the Internet. The proposed approach allows to prevent excessive consumption of electricity and overload of the power supply network under normalized (or limited in power) consumption of energy resources. The hardware / software system has developed on the Arduino platform with Bluetooth control.

Keywords: IoT block, Arduino, Bluetooth, power consumption limitation, non-priority device, automatized power unplug.

Вступ. Сучасні бездротові мережеві технології є ключовими для впровадження технології Internet-of-Things (IoT) в індустріальний сектор, створення апаратно-програмних ресурсів для розробки адаптивних алгоритмів споживання електроенергії. Широко використовуються стандарти протоколів Wi-Fi, Bluetooth, ZigBee та ін. найбільш ефективно забезпечують режими обміну між пристроями при невеликому розмірі переданих пакетів даних. Крім того, напрацьована достатня база сумісності декількох з названих технологій бездротового зв'язку в одній керуючій системі, що забезпечує раціональне споживання електроенергії [1].

Вбудована технологія індустріальної IoT (IIoT) дозволяє не тільки здійснювати віддалене керування приладами, а й вдосконалювати автоматизацію, керування даними в режимі реального часу та підключення приладів без втручання особисто споживача, але на основі інтелектуалізації сегментів IoT-мереж [2]. Тобто, технологія IoT спроможна сприяти досягненню економічно вигідного балансу між виробництвом енергії та попитом, впливати на процеси генерації, розподілу, споживання та зберігання енергії.

Окрім ринкового енергетичного менеджменту, також актуальним у багатьох країнах є підхід з державним регулюванням, головним інструментом якого є нормування споживання енергоресурсів [3].

Так, в Україні відповідно до п. 37 Правил користування електричною енергією для населення (надалі – ПКЕЕН), затверджених Постановою КМУ від 29 липня 1999 р. № 1357, енергопостачальник має право встановлювати технічні засоби, які обмежують постачання електричної енергії споживачу відповідно до договірної потужності, що становить для однофазних електроустановок побутових споживачів 2,2 кВт [4]. За додатковими договорами споживача з постачальником («Обленерго») у шафі обліку з лічильником електроенергії (які знаходяться на балансі електропередавальної організації) може бути встановлений автоматичний вимикач (АВ) на 16 А, який розрахований на номінальне навантаження 3,5 кВт.

Але сучасна квартира з бойлером (1–2 кВт), електрочайником (1–2,5 кВт), пральною машиною (1,7–2,9 кВт), СВЧ-пічкою (500–2500 Вт), індукційною плитою (3,2 кВт) потребує набагато більше електроенергії для одночасного споживання. Тому тими ж ПКЕЕН (п. 41) передбачена можливість встановлення споживачами у своєму приватному домогосподарстві генеруючої установки потужністю до 30 кВт.

З аналізу сучасного стану підключення побутових споживачів до міської електромережі зроблений висновок, що найпоширенішим є підключення на 5,5 кВт споживання (АВ 25 А на вході).

На жаль, більшість споживачів все ж мають дотримуватись дозволеною потужністю до 3,5 кВт одночасно включених пристроїв. Саме тут Інтернет речей (IoT) може мати найбільший вплив на спосіб розподілу та споживання електроенергії.

Термін «Інтернет речей» вперше був введений Кевіном Ештоном у 1999 р. під час його роботи над Procter & Gamble, щоб порахувати і відстежити товари без потреби в людському втручанні [5]. Підключення до глобальної мережі Інтернет при цьому не є обов'язковим. Тобто, IoT-мережа це мережа унікально ідентифікованих об'єктів, які здійснюють інтелектуальну взаємодію без людського втручання через IP-подібні з'єднання [6].

Таким чином, за концепцією IoT, мережа складається з взаємопов'язаних пристроїв, які мають можливість приведення в дію, функцію програмування та ідентифікації, а також дозволяють виключити необхідність участі людини за рахунок використання інтелектуальних інтерфейсів.

Тому в даній роботі розглядається актуальне питання автоматизованої взаємодії між комп'ютерною мережею та електромережею (англ. «Inter Net») шляхом встановлення з'єднання та обміну інформацією безпосередньо між побутовими приладами, без участі їх власника.

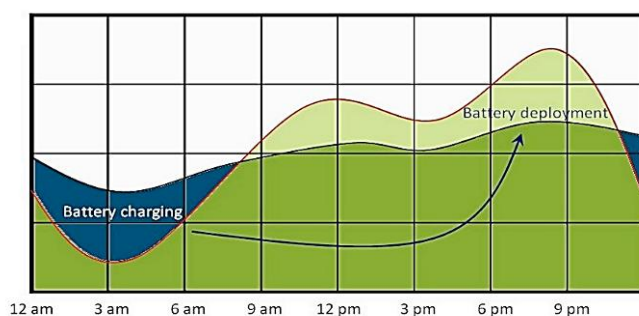
Постановка наукової проблеми. Метою даної роботи є створення апаратно-програмної системи запобігання надлишковому споживанню електроенергії та перенавантаженню мережі електроживлення при підключенні декількох приладів за допомогою технології IoT.

Аналіз останніх досліджень і публікацій.

Зараз на ринку є широкі можливості щодо реагування на попит, які впливають на підключені пристрої. Наприклад, на всій території США використовуються програмно-апаратні засоби для відключення пристроїв з високим споживанням електроенергії під час пікового попиту. Розумні термостати довели свою ефективність захисту від збоїв і відключень [7].

Крім того, на ринку з'явилися нові підключені пристрої, націлені на зміну енергокористування споживачами протягом доби або місяця.

Керування витратами на попит (англ. Demand Charge Management – DCM) можливо, наприклад, встановленням на будинок модульного накопичувача енергії Sparkplug Power [8]. У разі перевищення споживачем підведеної потужності, споживання переводиться з централізованої електромережі на літєві акумулятори (рис. 1).



а)



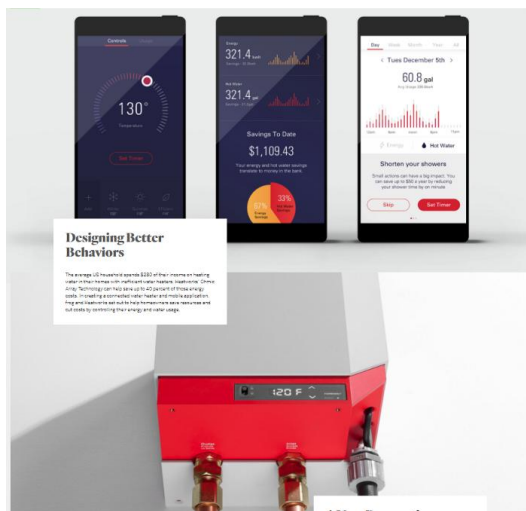
б)

Рис. 1. Керування витратами на попит за допомогою модульного накопичувача енергії Sparkplug Power

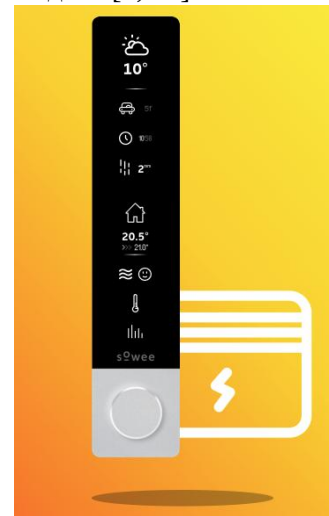
Але таке рішення підходить тільки для потужних споживачів, оскільки зазначене рішення орієнтоване на рівень споживання 250 кВт – 5 МВт.

На ринку з'являються все нові рішення, що дозволяють підключення та управління продуктами для виробництва, розподілу, управління та збереження енергії. Так, наприклад,

компанія Heatworks (США) у 2017 р. на виставці Consumer Electronics Show (CES – 2017) запропонував використовувати енергоефективний водонагрівач з підтримкою IoT на платформі Xively (рис. 2, а) – з повним контролем температури та виконанням завдань [9, 10].



а)



б)

Рис. 2. IoT-рішення з керування енергоспоживанням: а – фірми Heatworks (США); б) – фірми Soweel (Франція)

Іншим прикладом є французьке рішення – підключена домашня станція Soweel (рис. 2, б), яка також дозволяє споживачам встановлювати бюджет на місяць з коригуванням енергоспоживання на основі цього бюджету шляхом зниження температури нагріву води, температури «теплої підлоги» тощо) [11].

На ринку України присутні такі регулятори навантаження електромережі, як «AR-01», реле відключення неперіоритетних пристроїв ABB LSS, реле пріоритету «РІПН-1-25» та ін. [12–17].

Наприклад, автоматичний регулятор навантаження мережі являє собою пристрій з електронною схемою управління, виконаний в пластмасовому корпусі з мережевою вилкою і двома розетками (рис. 3, а). Одна з розеток має маркування «Priority». Робоча напруга 220 В, допустимий струм кожної з розеток 16 А. Під розетками розташовані світлодіодні індикатори, що показують режим роботи приладу [12]. Пристрій має фіксовані установки і не потребує будь-яких налаштувань, але передбачає підключення всіх керованих пристроїв тільки в одному приміщенні (рис. 3, б).



а)



б)

Рис. 3. Автоматичний регулятор навантаження мережі «AR-01»: а – зовнішній вигляд; б – варіант підключення [12]

Інші розповсюджені на вітчизняному ринку пристрої контролю споживаної потужності встановлюються в електричне коло після ввідного АВ перед навантаженням. Такі пристрої

здійснюють постійний контроль поточної споживаної потужності та порівнюють її із заданим максимально допустимим значенням. У разі перевищення загальної споживаної потужності заданого максимуму, реле відключає неперіоритетні навантаження, чим не допускає спрацьовування головного АВ.

Альтернативним виглядає рішення з використанням GSM-розетки або WiFi-розетки, які управляють подачею електроживлення вставленого в них приладу після команд, отриманих від власника через відповідну віддалену мережу [18]. «Розумна розетка» складається з контролера і електромагнітного реле. Після отримання команди контролер її розпізнає і передає відповідний сигнал реле, яка в свою чергу замикає або розмикає електричний ланцюг. Сигнали контролера можуть надходити за допомогою інтернету, SMS-повідомлень, по GSM каналам зв'язку.

Слід відмітити, що всі зазначені рішення не є адаптивними. При керуванні навантаженням, організованим у локальній мережі, розподіл на пріоритетне та неперіоритетне навантаження мережі користувача приймається одноразово та назавжди при виконання будівельно-монтажних робіт у приміщенні. В той же час, в процесі фахової або побутової діяльності у споживача можуть мінятися поточні пріоритети для різних пристроїв, навіть протягом доби. Але, пріоритети пристроїв у всіх зазначених рішеннях не можуть бути змінені власником пристроїв власноруч, а потребують при кожній такій зміні залучення фахівців енергокомпаній.

Навпаки, при тільки віддаленому типу керування електроспоживанням у локальній мережі остання знаходиться під постійною загрозою Інтернет-хакінгу. При такому типу керування ніякі підключення/відключення не виконуються автоматично, навіть такі, що є однотипними та не потребують змін режимів роботи.

Сучасний стан розвитку комп'ютерно-інтегрованих систем дозволяє переглянути жорстку однозначність у визначенні типу керування або пріоритетів підключень пристроїв відповідно до поточних потреб споживача. Із зростанням кількості розумних пристроїв, підключених за технологією IoT, з'являється можливість надавати споживачам (на мобільні або стаціонарні комп'ютерні системи) докладний вигляд того, які прилади використовують електроенергію в їх будинку у кожен момент часу. Крім того, можливо забезпечувати комбінування віддаленого та автоматизованого контролю цих пристроїв, щоб або зменшити споживання енергії, або попередити її відключення через перенавантаження електромережі споживача електроприладами.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Розробка та підключення IoT-блоків до кожного пристрою споживача дозволяє замість централізованої технології керування, структура якої обирається та реалізується одноразово, реалізувати адаптивні алгоритми роботи таких пристроїв з урахуванням поточних потреб споживача.

На рис. 4 наведені схематичні рішення IoT-блоків двох типів (пріоритетного та неперіоритетного навантаження) на платформі Arduino [19–21].

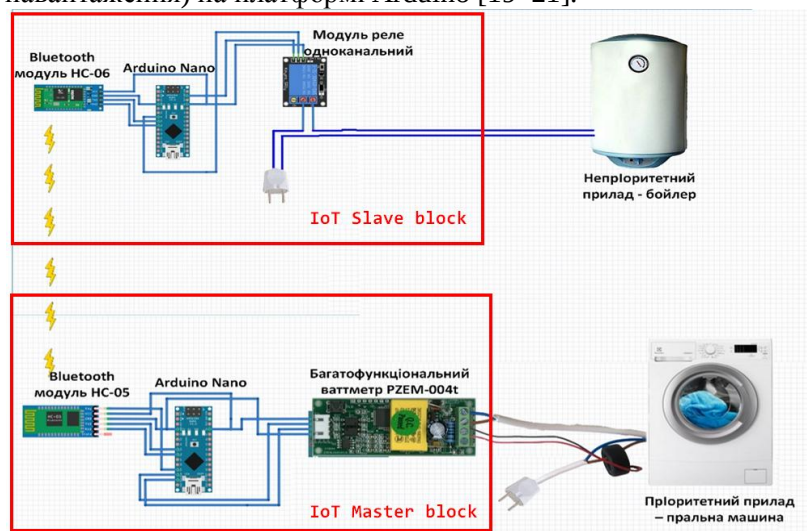


Рис. 4. Схема підключення модулів IoT-блоків пріоритетного та неперіоритетного побутових приладів

Починається робота блоків з їх підключення до електромережі та вибору користувачем пріоритетного та неперіоритетного приладу на певний термін. Далі стартують нескінченні (до

повного вимкнення блоків з електромережі) цикли роботи для кожного з IoT-блоків, блок-схема алгоритму якої наведена на рис. 5. Якщо почав працювати прилад, визнаний пріоритетним у даний час, то до непріоритетних IoT-блоків через Bluetooth-мережу відправляється сигнал на припинення роботи. Після вимкнення через певні інтервали часу йде перевірка входних сигналів на поновлення роботи: якщо такий сигнал прийшов від пріоритетного блоку – відбувається поновлення роботи та повертання на початок циклу, якщо ні – повертання вимкненого стану.

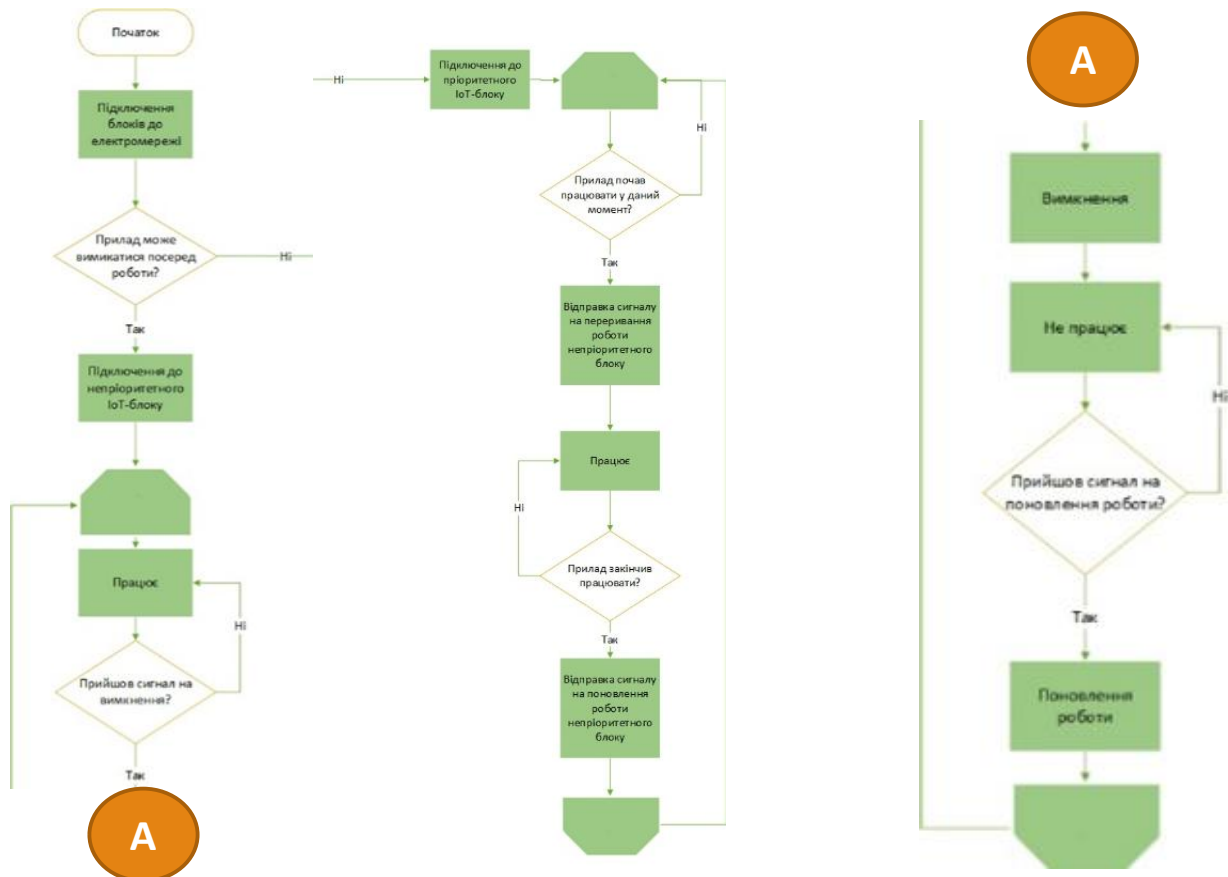


Рис. 6. Алгоритм роботи системи почергового підключення побутової техніки до електромережі

Кожен IoT-блок складається з чотирьох основних компонентів.

Пріоритетний (IoT Master block):

- плата Arduino Nano (ATmega328);
- Bluetooth-модуль для Arduino (HC-05 6 pin);
- багатофункціональний ватметр PZEM-004t;
- електричний подовжувач.

Непріоритетний (IoT Slave block):

- плата Arduino Nano (ATmega328);
- Bluetooth-модуль для Arduino (HC-06 4 pin);
- блок реле одноканальний 5 V;
- електричний подовжувач.

Такі блоки працюють в автоматизованому режимі відразу після розподілу на пріоритетне та непріоритетне навантаження в електромережі споживача. Віддалене керування не потрібне аж до моменту, коли необхідно змінити режим підключення приладів.

Також можна обладнати головний модуль можливістю керування через мережу Інтернет (рис. 7). Для цього у вже існуючу схему потрібно додати Wi-Fi модуль ESP8266 ESP-01, який буде приймати сигнали з мобільного пристрою споживача або зі стаціонарного серверу, де встановлюється відповідний програмний застосунок для керування описаною IoT-мережею.

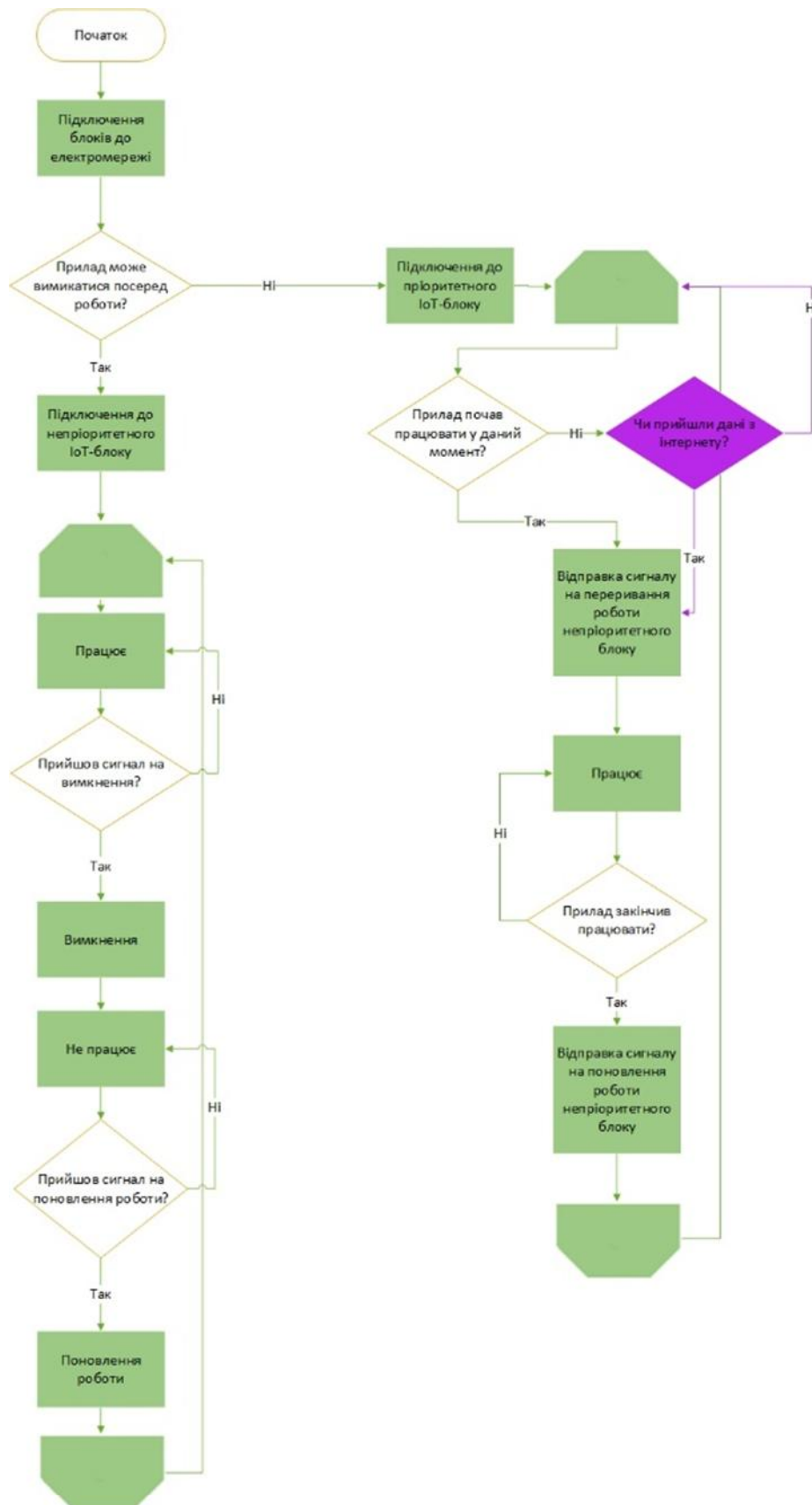


Рис. 7. Алгоритм роботи системи почергового підключення приладів до електромережі (з можливістю керувати через Інтернет)

У разі необхідності віддаленого керування через Інтернет наведений алгоритм буде змінено для пріоритетного IoT-блоку таким чином (рис. 7):

– перевірка, чи почав прилад працювати у даний час (якщо почав, то відправка сигналу до неперіоритетного IoT-блоку на припинення роботи; якщо не почав – то перевірка надходження команд через Інтернет);

– перевірка надходження даних з Інтернету (якщо прийшли, то відправка сигналу на припинення роботи до неперіоритетного IoT-блоку; якщо не прийшли – повернення на початок циклу);

– робота;

– перевірка, чи вимкнувся прилад (якщо так, то відправка сигналу на поновлення роботи до неперіоритетного IoT-блоку та перехід до початку циклу; якщо ні – повернення до роботи).

Висновки та перспективи подальших досліджень. В результаті роботи проаналізовано існуючі рішення щодо керування електроспоживанням побутових приладів через локальну мережу.

Розроблене апаратне рішення на платформі Arduino для відключення неперіоритетного електроприладу через локальну мережу при початку циклу роботи пріоритетного приладу. Розроблене програмне забезпечення (ПЗ), що дозволяє автоматизувати процес почергового підключення зазначених приладів до електромережі.

Розроблений алгоритм та структурна схема апаратного модулю для віддаленого керування розробленим пристроєм через Інтернет.

Проаналізовані можливості та принципи організації раціонального енергоспоживання на основі технології «Інтернету речей». Викладено теоретичні підходи до створення адаптивних алгоритмів оптимізації електроживлення в умовах обмеженої потужності.

Запропонований алгоритм автоматизованого почергового підключення приладів до мережі електроживлення шляхом розподілу на пріоритетне та неперіоритетне навантаження з використанням протоколу Bluetooth. Використання бездротових технологій суттєво розширює галузь застосування запропонованого підходу з керування підключенням приладів у приміщенні до керування підключенням до енергоспоживання територіально розподілених систем (у т. ч. безпілотних, без втручання людини). Передбачені два режими роботи: автоматизований у локальній IoT-мережі та з можливістю віддаленого керування.

Розроблена відповідна апаратно-програмна система на платформі Arduino. Розглянуто приклади реалізації IoT-мережі на базі Bluetooth-пристроїв для побудови системи раціонального споживання електроенергії.

Використання запропонованих рішень з інтеграції компонентів комп'ютерної та електричної мереж дозволяє запобігти перенавантаженням електромережі, забезпечити безперервність циклу роботи пріоритетних приладів та позбавитись від аварійних відключень всіх приладів споживача від електроживлення. Крім того, такий підхід суттєво підвищує безпеку впровадження IoT-рішень в локальній мережі через відсутність постійного підключення останньої до глобальної мережі Інтернет. Використання віддаленого керування здійснюється тільки у разі необхідності зміни запрограмованих заздалегідь режимів.

Напрямом подальшого дослідження є розробка способів збільшення відстані між IoT-блоками в приватній мережі та зменшення часової затримки при перепідключеннях апаратних компонентів.

1. Клевков А. В. Основы рационального потребления электроэнергии. – М. Инфра-инженерия, 2017. – 232 с.
2. Jadot F. Smart Control Harnesses the Industrial Internet of Things [Electronic resource] // Schneider Electric Blog. – Publ. June 30, 2016. – Available at: <https://blog.schneider-electric.com/machine-and-process-management/2016/06/30/smart-control-harnesses-industrial-internet-things/> (Last accessed: 14.03.2018).
3. Ганжа В. Л. Основы эффективного использования энергоресурсов: теория и практика энергосбережения. – Минск : Белорусская наука, 2007. – 451 с.
4. Правила користування електричною енергією для населення [Електронний ресурс] : затв. постановою КМУ від 29 липня 1999 р. № 1357. – Дата оновлення : 20.09.2016. – Режим доступу : <http://zakon0.rada.gov.ua/laws/show/1357-99-%D0%BF> (дата звернення 23.03.2018).
5. Ashton K. That 'Internet of Things' Thing [Electronic resource] // RFID Journal. – Publ. June 22, 2009. – Available at : <http://www.rfidjournal.com/articles/view?4986> (Last accessed : 19.03.2018).
6. McFarlane D. Auto ID systems and intelligent manufacturing control / D. McFarlane, S. Sarma, J. L. Chirn, C. Y. Wong, K. Ashton // Engineering Applications of Artificial Intelligence. – 2003, June. – Vol. 16, Is. 4. – June 2003, Pp. 365–376. doi: 10.1016/S0952-1976(03)00077-0.
7. The Internet of Things: Making the Energy Industry New (and Great) Again. Clean Technica [Electronic resource] : web-newspaper. – Publ. June 16, 2017. – Available at : <https://cleantechnica.com/2017/06/16/the-internet-of-things-making-the-energy-industry-new-and-great-again/> (Last accessed : 19.03.2018).

8. Reducing Electricity Costs for Commercial & Institutional Electric Customers with Energy Storage as a Service [Electronic resource]. – Available at: <http://sparkplugpower.com/wp-content/uploads/2015/04/SP-brochure-1APR2015.pdf> (Last accessed : 19.02.2018).
9. IoT for Smart Energy & Utilities. Xively IoT Platform [Electronic resource]: web-site. – Available at: <https://www.xively.com/navigating-iot/iot-for-smart-energy> (Last accessed : 19.03.2018).
10. The Heatworks team previewed their next-generation, IoT-connected MODEL 3 water heater [Electronic resource] // Proceedings of the Consumer Electronics Show (CES'2017), Las Vegas, USA, Jan. 8–12, 2017. – Available at: <https://www.frogdesign.com/work/heatworks> (Last accessed : 19.02.2018).
11. Je ne trouve pas le Wi-Fi Sowe dans la liste des réseaux Wi-Fi [Electronic resource]. – Available at: <https://monespace.sowe.fr/s/article/Je-ne-trouve-pas-le-wifi-Sowe-dans-la-liste-des-reseaux-wi-fi> (Last accessed : 19.03.2018).
12. AR-01 – Автоматический регулятор нагрузки сети [Электронный ресурс]. – Режим доступа : <http://ar01.real-el.ua/> (дата обращения 23.02.2018).
13. Как обеспечить комфорт при недостаточной мощности электросети [Электронный ресурс]. – Режим доступа : <http://elektrik.info/main/news/796-kak-obespechit-komfort-optimizator.html> (дата обращения 23.03.2018).
14. Умная розетка – Оптимизатор нагрузки на электросеть OEL-820 [Электронный ресурс]. – Режим доступа : <https://smartron.ru/catalog/products/saveenergy/umnaja-rozetka-optimizator-nagruzki-na> (дата обращения 23.03.2018).
15. Реле отключения неприоритетных нагрузок ABB LSS1/2 [Электронный ресурс]. – Режим доступа : <https://myvolt.ua/rele-otklyucheniya-neprioritetnykh-nagruzok-abb-lss12> (дата обращения 23.03.2018).
16. Реле приоритета нагрузок [Электронный ресурс]. – Режим доступа : <http://electricalschool.info/spravochnik/apparaty/1553-rele-prioriteta-nagruzok.html> (дата обращения 23.02.2018).
17. Реле приоритета. Автоматическое управление нагрузкой [Электронный ресурс] // Режим доступа : <http://elektrik-sam.info/rele-prioriteta-avtomaticheskoe-upravlenie-nagruzkoj> (дата обращения 23.02.2018).
18. GSM-розетки [Электронный ресурс]: интернет-магазин. – Режим доступа : https://ohrana.ua/gsm-signalizatsiya/gsm-rozetka/?gclid=EAIaIQobChMlrMW2tqmG2gIVjZztCh3zfgU4EAAAYASAAEgI2P_D_BwE (дата обращения : 18.03.2018).
19. Arduino Nano (V3.0): User Manual [Electronic Resource]. – Available at: http://site.gravitech.us/Arduino/NANO30/Arduino_Nano3_0.pdf (Last accessed: 14.03.2018).
20. Ардуино в Украине [Электронный ресурс]: интернет-магазин. – Режим доступа : <https://arduino.ua/> (дата обращения : 18.03.2018).
21. Модуль Bluetooth HC-05 [Электронный ресурс] // Arduino-KiT : веб-сайт. – Режим доступа : <https://arduino-kit.ru/catalog/id/modul-Bluetooth-hc-05> (дата обращения : 18.03.2018).

UDC 004.93.1; 681.142

^{1,2}Заяць В.М., д.т.н., професор; ¹Маєвський Яцек, PhD, доцент; ¹Марціняк Томаш, PhD, доцент;
³Рибицька Ольга Мар'янівна, к.фіз.-мат.н., доцент; ³Заяць Марія Михайлівна, ст. викладач.
¹Університет Технологічно-Природничий (УТР), м. Бидгощ, Польща.
²Національний університету водного господарства та природокористування, м. Львів, Україна.
³НУ "Львівська політехніка", м. Львів, Україна.

MODELS AND ALGORITHMS FOR PROCESSING OF FUZZY INFORMATION

Zaiats V.M., Majewski J., Marciniak T., Rybyska O.M., Zaiats M.M. Models and algorithms for processing of fuzzy information. Methods and algorithms for the solution of applied problems of processing of statistical information on the basis of the theory of fuzzy integrals and blurry logic are proposed. The developed approaches provide a good choice of belonging function, the criterion of likelihood and fuzzy measure based on the knowledge and intuition of the expert. It is substantiated the expediency of using declarative programming languages for the processing of statistical information.

Keywords: fuzzy information; fuzzy set; blurry logic; belonging function; criterion of likelihood; declarative programming languages.

Заяць В.М., Маєвський Я., Марціняк Т., Рибицька О.М., Заяць М.М. Моделі та алгоритми для опрацювання нечіткої інформації. Запропоновано методи та алгоритми до розв'язання прикладних задач обробки нечіткої інформації на основі теорії нечітких інтегралів та розмитої логіки. Розроблені підходи забезпечують вдалий вибір функції приналежності, критерію ймовірності та нечіткої міри, заснованої на знанні та інтуїції експерта. Обґрунтовується доцільність використання декларативних мов програмування для обробки статистичної інформації.

Ключові слова: нечітка інформація; нечітка множина; розмита логіка; функція приналежності; критерій вірогідності; декларативні мови програмування.

Заяць В.М., Маєвський Я., Марціняк Т., Рыбыцка О.М., Заяць М.М. Модели и алгоритмы для обработки нечеткой информации. Предложены методы и алгоритмы к решению прикладных задач обработки размытой информации на основе теории нечетких интегралов и размытой логики. Разработанные подходы обеспечивают удачный выбор функции принадлежности, критерия вероятности и нечеткой меры, основанной на знании и интуиции эксперта. Обосновывается целесообразность использования декларативных языков программирования для обработки статистической информации.

Ключевые слова: нечеткая информация; нечеткое множество; размыта логика; функция принадлежности; критерий достоверности; декларативные языки программирования.

I. Formulation of the problem. The problems of mathematical and computer modeling today lie, in particular, in the inappropriateness of clear logic and task model with clearly defined input parameters in cases where, for some reason, there are contradictions, uncertainty or lack of clarity of information about the object under study, system or phenomenon [1, 4, 6, 8, 11, 13-17].

Uncertainty is known to arise because of the lack of knowledge relating to a particular event [2]. She is present to the experiment. Mathematical models for information uncertainty objects are based on the probability theory, the theory of opportunities, measures of confidence, fuzzy logic and several others.

The phenomenon of fuzziness arises in the process of combining into one whole of objects, which have a common property:

$$X = \{x | \text{owns } \varphi\}$$

where x - all the elements of a certain universal set.

Given that there are always elements in the reality that it is unclear whether they possess the specified property or not, it is not a plural in the classical sense. Any attempt to interpret the general description leads to fuzzy concepts, since the exact description contains an excess of details. Increasing the accuracy of the description leads to an increase in the amount of information, the content of which decreases until the time when the accuracy and meaningfulness do not become mutually exclusive. For the first time, L.A. Zadeh [3] stressed the need for uncertainty for the transmission of content. It was the ideas of this American scientist give pushed for the development of "fuzzy mathematics" [4], which, along with the apparatus of fuzzy sets, contains other methods of work with uncertainty.

The application of the theory of fuzzy sets and measures is a step towards the convergence of the precision of classical mathematics with a false inaccuracy of the environment, an attempt to overcome the linguistic barrier between a person whose judgment and evaluation are approximate and fuzzy with technical means (by software), which can only carry out precise instructions [5].

II. Analyses of last researches and aim of work. A device that allows you to work with fuzzy logic, blurred" parameters of models, is a Fuzzy-technology device [1-3, 8]. The Fuzzy-Technology division has some kind of expert systems [5, 6].

Fuzzy output systems are designed to implement the fuzzy output process and serve as a conceptual basis for all modern fuzzy logic. Fuzzy output systems allow you to solve the tasks of automated control, data classification, image recognition, decision-making, and many others.

Linguistic variables extend the ability to represent knowledge. They are determined by fuzzy sets, whose values are determined by functions of fidelity. Affiliation functions can be obtained through expert assessments [6], or by analysing fuzzy clusters. According to [6], fuzzy expert systems can be implemented when the cost of obtaining accurate information, that is, information in absolute terms, exceeds the maximum revenue from the restructuring of a model or is practically impossible.

It is known that the initial stage of constructing artificial intelligence on the basis of the use of natural language is based on ambiguous logic and the mechanism of output with rigid rules. First-generation expert systems work on the following set of rules:

*If X it belongs A_k , then Y_k it belongs C_k
 were $k = 1, \dots, N$.*

Here, there is a set of exact numerical or symbolic values that can be linguistic meanings of linguistic variables.

At present, in expert systems, productive models are the most used formalism for representing knowledge

"If A , then B ".

The second generation of expert systems has at least two peculiarities: fuzzy representation of knowledge and fuzzy calculations. One of the most common tasks of logical output in the conditions of fuzziness can be formulated as follows

*Given (fuzzy) production rule "If A then B ".
 Observed A (A to some extent). What should be B ?*

After obtaining a fuzzy set of conclusions, we find a specific numerical means (conducting dephasing).

The purpose of this work is to develop a fuzzy data analysis algorithm, which consists of creating a fuzzy set of data model, creating a fuzzy knowledge data base model and selecting the membership functions for the purpose of effective classification of data and their statistical processing.

III. Building models with blurry logic. Consider an object with one output and type inputs

$$y = f(x_1, x_2, \dots, x_n), \quad (1)$$

where $x_1 \dots x_n$ - the set of values of input variables; y - output variable.

For construct a mathematical model on the basis of establishing the relationship between input and output variables in accordance with experimental data we shell make the operation of pacification and quantitative and qualitative variables are translated into linguistic terms.

$$U_i = [\underline{u_i}, \overline{u_i}], i = \overline{1, n} \quad (2)$$

$$Y = [\underline{y}, \overline{y}] \quad (3)$$

where $\underline{u_i}, \overline{u_i}$ - the smallest and the highest possible value y of the variable x_i ; $\underline{y}, \overline{y}$ - the smallest and the highest possible value of the output variable y .

To solve the problem (1) it is necessary to apply a method of making a decision by means of which a fixed vector of input variables $x^* = (x_1^*, x_2^*, \dots, x_n^*)$, $x_i^* \in U_i$ would unambiguously be placed in accordance with the solution $y^* \in Y$. For the formal establishment of this dependence we shall consider the input x_i , $i = \overline{1, n}$ and output parameters y as linguistic variables given on universal sets (2), (3). To evaluate linguistic variables x_i , $i = \overline{1, n}$, and y we will use qualitative terms from the following term sets: $A_i = \{a_i^1, a_i^2, \dots, a_i^{p_i}\}$ – term - set of input variable x_i , $i = \overline{1, n}$; $D = \{d_1, d_2, \dots, d_m\}$ – term -set of output variable y .

For the construction of term -sets, one can apply, for example, the method proposed in [16].

For each term of each linguistic variable, based on expert knowledge, the functions of membership (trapezoidal, triangular, rectangular, sinusoidal, parabolic, etc.) [5] are constructed based on expert knowledge, where $\mu^{a_i^p}(x)$ – the degree of belonging of the element $x \in U_i$ to the term $a_i^p \in A_i$, $i = \overline{1, n}$; $p = \overline{1, p_i}$; $\mu^{d_j}(y)$ – of the measure of the membership of the element y to the term $d_j \in D$, $j = \overline{1, m}$.

The definition of linguistic estimates of variables and the functions of membership necessary for their formalization is the first stage in the construction of a fuzzy model of the object being studied. In the literature on fuzzy logic, he received the name of the phasing of variables (English fuzzification) [8].

IV. Approach to building Models of Fuzzy Knowledge Bases. The next step is to create a base of fuzzy knowledge. Let the object (8) know the rules that connect its inputs and output using vectors such as:

$$N = k_1 + \dots + k_j + \dots + k_m, \quad (4)$$

where k_j – the number of experimental data corresponding to the same value d_j of the term-set of the output variable y ; m – the total number of terms of the output variable, and in the general case $k_1 \neq \dots \neq k_m$.

We will assume that the number of available experimental data $N < p_1 p_2 \dots p_n$, is less than the total overview of various combinations of possible terms of input variables p_i , $i = \overline{1, n}$. Then the knowledge base is a table formed according to the following rules:

1. The table's size $N \times (n+2)$, is equal to the number of columns $n+2$ – and N – the number of rows;
2. Each row of the matrix – a combination of input variables assigned by the expert to one of the possible values of the term-set of the output variable y . In this case, the first k_1 lines correspond to the value of the output variable $y = d_1$, the following k_2 lines – the value $y = d_2$, etc., and the last k_m lines – the value $y = d_m$;
3. The first n columns of the matrix correspond to the input variables x_i , $i = \overline{1, n}$; $(n+1)$ -st – the weight of the rule W_{jp} , $j = \overline{1, m}$, $p = \overline{1, k_j}$, $(n+2)$ -nd – the value d_j of the output term-set of the variable y , $j = \overline{1, m}$, corresponding to the combination of values in the first $(n+1)$ columns.
4. The element located at the intersection of the i -th column and the jp -line corresponds to the linguistic evaluation of the parameter x_i in the row of knowledge matrix with the number jp . In this case, the linguistic assessment a_i^{jp} is chosen with the term-set corresponding to the variable x_i , i.e. $a_i^{jp} \in A_i$, $i = \overline{1, n}$; $j = \overline{1, m}$; $p = \overline{1, k_j}$.

Table 1. General view of the fuzzy knowledge base

Number Incoming combination	Input variables				Weight	Output variable
	x_1	x_2	$\dots x_i \dots$	x_n	w	y
11	a_1^{11}	a_2^{11}	a_i^{11}	a_n^{11}	w_{11}	d_1
12	a_1^{12}	a_2^{12}	a_i^{12}	a_n^{12}	w_{12}	
...	...	...	...	...	...	
$1k_1$	$a_1^{1k_1}$	$a_2^{1k_1}$	$a_i^{1k_1}$	$a_n^{1k_1}$	w_{1k_1}	
...	...	...	...	...	...	...
j_1	a_1^{j1}	a_2^{j1}	a_i^{j1}	a_n^{j1}	w_{j1}	d_j
j_2	a_1^{j2}	a_2^{j2}	a_i^{j2}	a_n^{j2}	w_{j2}	
...	...	...	...	...	...	
jk_j	$a_1^{jk_j}$	$a_2^{jk_j}$	$a_i^{jk_j}$	$a_n^{jk_j}$	w_{jk_j}	
...	...	...	...	...	...	...
m_1	a_1^{m1}	a_2^{m1}	a_i^{m1}	a_n^{m1}	w_{m1}	d_m
m_2	a_1^{m2}	a_2^{m2}	a_i^{m2}	a_n^{m2}	w_{m2}	
...	...	...	...	...	...	
mk_m	$a_1^{mk_m}$	$a_2^{mk_m}$	$a_i^{mk_m}$	$a_n^{mk_m}$	w_{mk_m}	

When an expert creates linguistic rules such as "IF THING" that form the basis of fuzzy knowledge about a particular object, the expert's confidence in each rule may be different. If one rule in the opinion of an expert can serve as an undeniable truth, then according to another rule in the same expert there may be some doubts.

In order to reflect these different degrees of confidence in the base of fuzzy knowledge, the weighting of the rules is introduced - these are numbers from the interval [0, 1], which characterize the expert's confidence in each specific rule chosen by him to make a decision. The general view of the knowledge base is given in Table. 1.

After building the knowledge base, you need to carefully check in Table 1 the presence of opposite lines of content, that is, rules that the same input variables have different output values. The introduced

matrix of knowledge defines a system of logical utterances such as "YES - THEN, OTHER", which associate the values of input variables with one of the possible output values $d_j, j = \overline{1, m}$:

IF $(x_1 = a_1^{11})$ AND $(x_2 = a_2^{11})$ AND ... AND $(x_n = a_n^{11})$ (with weight w_{11}),
 OR $(x_1 = a_1^{12})$ AND $(x_2 = a_2^{12})$ AND ... AND $(x_n = a_n^{12})$ (with weight w_{12}),
 OR ...
 AND $(x_1 = a_1^{1k_1})$ AND $(x_2 = a_2^{1k_1})$ AND ... AND $(x_n = a_n^{1k_1})$ (with weight w_{1k_1}),
 THEN $y = d_1$, OTHER
 IF $(x_1 = a_1^{21})$ AND $(x_2 = a_2^{21})$ AND ... AND $(x_n = a_n^{21})$ (with weight w_{21}),
 OR
 OR $(x_1 = a_1^{2k_2})$ AND $(x_2 = a_2^{2k_2})$ AND ... AND $(x_n = a_n^{2k_2})$ (with weight w_{2k_2}),
 THEN $y = d_2$, OTHER ...
 IF $(x_1 = a_1^{m1})$ AND $(x_2 = a_2^{m1})$ AND ... AND $(x_n = a_n^{m1})$ (with weight w_{m1}),
 OR
 OR $(x_1 = a_1^{mk_m})$ AND $(x_2 = a_2^{mk_m})$ AND ... AND $(x_n = a_n^{mk_m})$ (with weight w_{mk_m}), THEN $y = d_m$.

A similar system of logical expressions is called a fuzzy knowledge base. Using the operations $\cup$ (OR) and $\cap$ (AND) described system of logical statements can be rewritten in a more compact form:

$$\bigcup_{p=1}^{k_j} \left[\bigcap_{i=1}^n (x_i = a_i^{jp}) \right] \rightarrow y = d_j, j = \overline{1, m}. \quad (5)$$

Thus, the input relation (1), which establishes the connection between the input parameters and the output variable, is formalized in the form of a system of fuzzy logical statements (5) based on the created matrix of knowledge. The rules for fuzzy considered of the described system, in which the degree of truth is different from zero, named active.

In [10], a method is proposed to use fuzzy logic equations that are based on a knowledge matrix or isomorphic system of logical expressions (5) and allow us to calculate the values of the membership function of the output variable for the fixed values of the inputs of the object.

Linguistic estimates a_i^{jp} of the variables $x_1, \dots, x_n$, contained in the logical statements (5) will be considered as fuzzy sets defined on universal sets (2). We introduce the following notation:

$\mu^{a_i^{jp}}(x_i)$ – function-membership of parameter x_i to fuzzy term a_i^{jp} , $i = \overline{1, n}$, $j = \overline{1, m}$, $p = \overline{1, k_j}$; $\mu^{d_j}(x_1, x_2, \dots, x_n)$ – function-membership vector of input variables $x = (x_1, x_2, \dots, x_n)$ to term of the output variable $y = d_j$, $j = \overline{1, m}$.

Thus, we have two types of functions, the relationship between which is determined by the base of fuzzy knowledge (5), on the basis of which you can output a system of logical equations, which can be submitted in a compact form:

$$\mu^{d_j}(x_1, x_2, \dots, x_n) = \bigvee_{p=1}^{k_j} \left(w_{jp} \left[\bigwedge_{i=1}^n \mu^{a_i^{jp}}(x_i) \right] \right), j = \overline{1, m}; \quad (6)$$

where $\vee$ – logical «OR»; $\wedge$ – logical «I».

The decision $d^* \in D\{d_1, d_2, \dots, d_m\}$, that corresponds to a fixed vector of the values of input variables $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ will be carried out in accordance with the following algorithm constructed using the apparatus of the fuzzy (blurry) logic [10]:

1. The possible range of change of controlled parameters will determined, a knowledge base is created with the use of expert data and a system of fuzzy logic equations is derived (6).

2. The vector of the values of the input variables $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ is fixed.

3. We specifies the function - membership of fuzzy term for different controlled parameters.

4. Using logical equations (6), the values of many parametric functions membership $\mu^{d_j}(x_1^*, x_2^*, \dots, x_n^*)$ of vector x for all values $d_j, j = \overline{1, m}$ of the output variable y are calculated. In this case, the logical operations $\cup$ (OR) and $\cap$ (AND) over the membership functions are replaced by the operations max and min:

$$\mu(a) \vee \mu(b) = \max[\mu(a), \mu(b)], \quad (7)$$

$$\mu(a) \wedge \mu(b) = \min[\mu(a), \mu(b)]. \quad (8)$$

That is, first find the minimum values of belonging functions in each rule, and then among them they choose the highest value of the membership function among all rules for each value, which corresponds to the original variable y . Thus, the conclusion is made that the origin variable belongs y to a term d_j^* , whose membership function is maximal.

The proposed algorithm uses the idea of identifying the linguistic term by the maximum of membership function and generalizes this approach to the entire knowledge matrix. The computational part of this algorithm is easily realized by simply applying operations max and min.

To obtain a clear number $[y, \bar{y}]$ from an interval that corresponds to a fuzzy value of the output variable, it is necessary to apply a dephasing operation. This clear number y^* can be determined, for example, by the center of gravity method:

$$y^* = \frac{\int_{Min}^{Max} y \mu(y) dy}{\int_{Min}^{Max} \mu(y) dy} \quad (9)$$

Obviously, to implement the described approach to the processing of fuzzy information it is advisable to use declarative programming languages. These include Lisp, Prolog or their modifications depending on a specific objective problem) [13-15], since they are most successfully adapted for the realization of functions of the form (2), (3), (5) - (9) which can be both analytic, so and descriptive (functional, logic rules, fuzzy sets). Such software allows solving tasks related to qualitative recognition and analysis of complex structures (handwriting recognition, handwriting, psychophysiological state of the person, construction and analysis of storage systems, processing, information security, automated proofing of the theorems, environmental monitoring and decision-making, social protection and economic development).

V. Results of numerical Analysis. Note that results computer modeling and numerical experiments a proud efficient applied described methods and algorithms to analysis fuzzy information.

In particular, the task of analyzing the efficiency of the pharmaceutical turnover of a pharmaceutical company under conditions of limited financing and short terms on the supply of goods

was considered. This approach proved to be effective, since it allows not only to minimize company expenses, but also to track the dynamics of turnover throughout the operating period.

One can propose a different approach to processing large volumes of fuzzy data under conditions of incomplete certainty of the vector of input variables (primary characteristics). The essence of the approach is based on conducting a simulation of the behavior of the investigated system and an expert assessment of the complementation of the existing knowledge base with new informative data and the establishment of the vector of input characteristics. Obviously, such an approach is iterative and it is necessary to take care of the convergence of the calculation process to achieve the goal with minimal cost and limited error.

When dealing with non-physical data in artificial intelligence problems, the construction of recognition systems, expert systems, medical and parametric diagnostics, the creation of logical-linguistic models, the most successfully adapted declarative programming languages. The logical statements and functional-logical dependencies provide the opportunity to describe the problem with fuzzy formulated data and obtaining solutions in the form of logical sequences, new functional dependences or probabilistic characteristics with definite mathematical hope and dispersion of the input signal. Ultimately, the initial vector of primary attributes should be refined, which will ensure reliable processing of fuzzy data.

VI. Conclusion and perspective forward researches. In the work, the authors proposed approaches to the processing of fuzzy information in the conditions of incomplete definition of the vector of input characteristics, which are based on the theory of fuzzy sets and measures, the construction of membership functions and the application of languages declarative programming.

The further development of the proposed approaches can be achieved by conducting statistical studies of specific application problems related to the need to assess both the quantity and value of information based on a figurative approach and theories of fuzzy sets.

The proposed approach is advisable to use in applied problems whose mathematical description is difficult or completely impossible. This approach will contribute to the development, in fact, of the methods of recognition theory and identification, processing of blurry data and the theory of information and coding.

1. Ueno X. *Introduction to knowledge engineering*. Tokyo: Omsya.- 1985. – 360 p.
2. Novak V., Perfiljeva I., Mochkorzh I., *Mathematical Principles of Fuzzy Logic*. M: Fizmalit, 2006. - 347 p.
3. Zadeh L.A. "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes". IEEE. Trans. Syst. Man and Cybern, 1, 1973. - P. 28-44.
4. Konyshova L.K., Nazarov D.M. *Fundamentals of the theory of fuzzy sets*. M: Peter, 2011. - 190 p.
5. Syvalko M. *Mathematics of hidden possibilities*. Ostrog: Publishing House of the National University "Ostroh Academy", 2011. - 394 p.
6. Turksen I.B. *Fuzzy expert systems* / Ed. M. Zheleny - St. Petersburg: Peter, 2002. - 1120 p.
7. Shapiro D.I., Blishchun A.F. "Choosing solutions with fuzzy state of the system". Algorithms and Programs, 1978, No. 1. - p.75.
8. Zimmermann H.J. *Fuzzy Set Theory and its Applications*. Kluner: Academic Publishers, Dordrecht, Boston, MA, 1991. - 315 p.
9. Dubois D. *Theory of Opportunity. Appendices to the representation of knowledge in computer science*. M.: Radio and Communication, 1990. - 288 p.
10. Rothstein O.P., Zhupanova M.O., Sheverda V.M. "Differential diagnosis of ischemic heart disease based on fuzzy logic". Bulletin of the Ministry of Education and Science, No. 3, 1994. - P. 32-40.
11. Kryukov A.P., Rodionov A.Ya. *The language of analytical computing*. Moscow: From Moscow State University, 1984. - 176 p.
12. Dyakonov V. *Matcad 2001*. SPB: Peter, 2001. - 624 pp.
13. Zaiats V.M. *Function programming*. Lviv: Publisher "Beskyd Beat", 2003. - 160 p.
14. C.In., D. Solomon. *Programming in Turbo Prolog*. M.: Svit, 1996. - 640 p.
15. Zaiats V.M., Zaiats M.M. Logical and functional programming. Tutorial. The Ministry of Education and Science of Ukraine (the list №1 / 11-16135 for 13 October 2014) provide Griff. Kamyanets-Podilsky: Publisher "Ruta". 2016. ISBN 978-617-738-138-8. - 400 p. (in Ukraine).
16. Zaiats V.M., Rybitska O. M The hidden possibilities of mathematics for the statistical processing of information // Materials International scientific and technical conf. "System Analysis and Information Technologies" (SAIT-2013), May 27-31, 2013, NTUU "KPI". Kiev, 2013. - P. 317-318.

УДК 681.515.8

Здолбіцька Н.В., Мельник Д.С.

Луцький національний технічний університет

СТВОРЕННЯ МОБІЛЬНОГО ДОДАТКУ В IDE ANDROID STUDIO ДЛЯ ДІАГНОСТИКИ СМАРТФОНІВ НА ОСНОВІ ОПЕРАЦІЙНОЇ СИСТЕМИ ANDROID

Здолбіцька Н.В., Мельник Д.С. Створення мобільного додатку в IDE Android Studio для діагностики смартфонів на основі операційної системи Android. Смартфони стали невід'ємною частиною повсякденного життя людей. Даний додаток дасть можливість отримати реальні дані про смартфон при покупці, а також продіагностувати поточний стан вашого смартфона. У даній статті розглядається розробка додатку в IDE Android Studio шляхом отримання системних даних з пристрою.

Ключові слова: смартфон, OS Android, IDE Android Studio, діагностика.

Здолбическая Н.В., Мельник Д.С. Создание мобильного приложения в IDE Android Studio для диагностики смартфонов на основе операционной системы Android. Смартфоны стали неотъемлемой частью повседневной жизни людей. Данное приложение позволит получить реальные данные о смартфоне при покупке, а также продиагностировать текущее состояние вашего смартфона. В данной статье рассматривается разработка приложения в IDE Android Studio путем получения системных данных с устройства.

Ключевые слова: смартфон, OS Android, IDE Android Studio, диагностика.

Zdolbitska N.V., Melnyk D.S. Research FIR filters using the LabVIEW environment. Create a mobile application in the Android Studio IDE to diagnose smartphones based on the Android OS. Smartphones have become an integral part of everyday life of people. This application will give you an opportunity to get real data about a smartphone when buying, and also to diagnose the current state of your smartphone. This article discusses the development of an application in the Android Studio IDE by obtaining system data from the device.

Keywords: smartphone, OS Android, IDE Android Studio, diagnostics.

Постановка наукової проблеми. Смартфони стали невід'ємною частиною повсякденного життя людей. Завдяки своїй портативності та великому спектру можливостей користувачі надають все більшу перевагу покупці смартфонів ніж комп'ютерів. Набагато простіше і швидше знайти якусь інформацію, відправити повідомлення, подивитись відеоролик на смартфоні аніж на комп'ютері. Тому дана галузь дуже стрімко розвивається, виробники все швидше впроваджують нові технології та покращують свої продукти. Купуючи смартфон, недобросовісний продавець може видати бракований чи гірший по характеристикам товар за завищеною ціною. Даний додаток дозволить зменшити зони ризику для покупця.

Аналіз досліджень. Середовище розробки Android Studio дозволяє швидко за допомогою різноманітних інструментів та можливостей мови програмування Java створювати мобільні додатки на основі операційної системи Android. Оскільки операційна система Android має відкритий код, розробник має доступ до системних даних та можливість використовувати їх. Також слід відзначити, що Android є найбільш популярною ОС на сьогоднішній день, близько на 60% смартфонів встановлена дана ОС. Це сприяє розвитку платформи та подальшому її поширенню, тому і розробка додатків для ОС Android є доволі популярною та прибутковою.

На сайті developer.android.com/develop можна знайти інформацію про те як працювати з різними класами, їх методами. Даний сайт містить всю необхідну та найактуальнішу документацію.

Аналізуючи Play Market можна зробити висновок, що додатки для діагностики смартфонів мають від 100 тисяч завантажень, що підтверджує необхідність того, що люди все частіше при покупці своїх девайсів, особливо тих, що вже були у використанні використовують дані додатки.

Виклад основного матеріалу та обґрунтування результатів дослідження.

Створюючи проект розробник може обрати найновішу версію Android, використовувати усі найновіші функції. Але у даному випадку у користувачів, які мають нижчі версії Android можуть бути проблеми у використанні додатку, або взагалі неможливість його використання у зв'язку із не підтримкою певних функцій. Даний вибір є не рентабельним з комерційної точки зору, оскільки зазвичай найновіші версії Android встановлено на 1-10% смартфонів. До прикладу Android 7.0 (Nougat), згідно з інформацією наданою в Android Studio встановлено 8.1% девайсів, старішу версію Android 6.0 (Marshmallow) - 39.3%. В той же час обравши нижчу версію ОС розробник не має можливості використовувати всі переваги нових версій, але підтримувати нормальну роботу додатку буде більша кількість смартфонів.

Даний проект створений для Android 5.0, оскільки 71% девайсів підтримуються з цієї версії Android, незважаючи на те, що дана версія не є найновішою, але ще доволі популярна у використанні.

Додаток складається з двох вкладок: Інформація (рис. 1) та Тести (рис. 2). При запуску відкрита вкладка Інформація, на якій зображена інформація про модель смартфона, та кнопки для переходу до інших Activity, на яких відображена інформація відповідно поділена по групах.

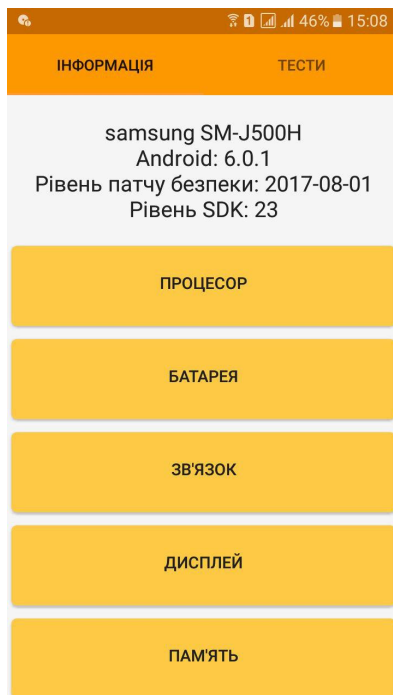


Рис. 1. Головний екран, вкладка Інформація

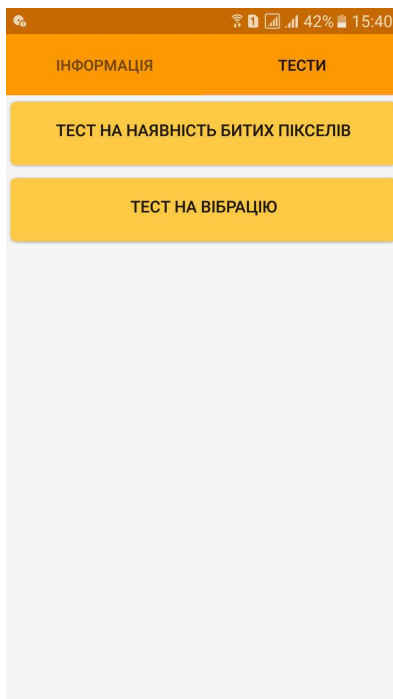


Рис. 2. Головний екран, вкладка Тести

Activity – це компонент додатка, який видає екран, і з яким користувачі можуть взаємодіяти для виконання будь-яких дій, наприклад набрати номер телефону, зробити фото, відправити лист або переглянути карту. Кожній операції присвоюється вікно для промальовування відповідного

призначеного для користувача інтерфейсу. Зазвичай вікно відображається на весь екран, проте його розмір може бути менше, і воно може розміщуватися поверх інших вікон.

Як правило, програма містить кілька операцій, які слабо пов'язані один з одним. Зазвичай одна з операцій в додатку позначається як «основна», пропонується користувачеві при першому запуску програми. У свою чергу, кожна операція може запустити іншу операцію для виконання різних дій. Кожен раз, коли запускається нова операція, попередня операція зупиняється, однак система зберігає її в стек («стек переходів назад»). При запуску нової операції вона поміщається в стек переходів назад і відображається для користувача. Стэк переходів назад працює за принципом LIFO (last in – first out) «останнім увійшов – першим вийшов», тому після того як користувач завершив поточну операцію і натиснув кнопку Назад, поточна операція видаляється з стека (і знищується), і відновлюється попередня операція. Саме так відбувається перехід по екранам (Activity), натиснувши кнопку назад по стандарту додаток згорнеться, і можна перейти на екран меню свого смартфона. В розробленому додатку при натисненні кнопки «Назад» на головному екрані, можна побачити сповіщення, яке повідомить: «Натисніть ще раз для виходу» (рис. 3). Реалізовано це наступним чином.

```
private static long back_pressed = 0;

@Override
public void onBackPressed()
{
    if (back_pressed + 2000 > System.currentTimeMillis())
        finish();
    else
        Toast.makeText(getBaseContext(), "Натисніть ще раз для виходу",
            Toast.LENGTH_SHORT).show();
    back_pressed = System.currentTimeMillis();
}
```

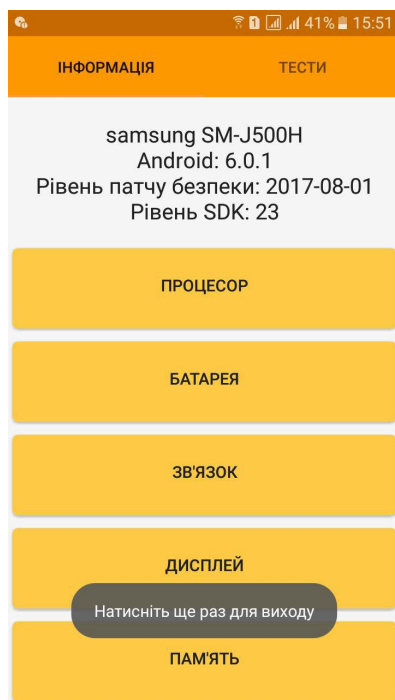


Рис. 3. Повідомлення при натисненні кнопки «Назад»

Деякі дані, такі як частота та використання процесора чи дані про оперативну пам'ять потребують постійного оновлення, для зображення поточних даних. Оновлення даних, наприкладі зображення частоти та використання процесора було реалізовано наступним чином. В методі *onCreate()* запускається метод *start()*, який запускає таймер, в свою чергу таймер кожної секунди запускає метод *get_freq_and_use()*, даний метод зчитує та подає на екран дані про поточну частоту та використання процесора.

```
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.cpu);
    cpuV = (TextView) findViewById(R.id.cpuV);
    freq_and_use = (TextView) findViewById(R.id.freq_and_use);
    header = (TextView) findViewById(R.id.cpu_header);
    header.setText("Процесор");
    get_freq_and_use();
    cpuV.setText("");
    cpuV.append(new Func().getCPUinfo());
    start();
}

protected void start()
{
    CountDownTimer countDownTimer = new CountDownTimer(seconds * 1000, 1000)
    {
        @Override
        public void onTick(long l) {
            get_freq_and_use();
            seconds++;
        }
        @Override
        public void onFinish() {
        }
    }.start();
}
```

Для отримання частоти використовувався клас *Frequency*. Об'єкт цього класу має методи *getCPUFrequencyCurrent()*, *getCPUFrequencyMin()* та *getCPUFrequencyMax()*. Дані про використання можна знайти у системному файлі */proc/stat*, інші дані про процесор отримано з файлу */proc/cpuinfo*.

Дані про батарею можна отримати з допомогою *BatteryManger*, вони теж потребують свого оновлення, але на відмінну від попереднього прикладу, оновлення інформації про батарею не потрібно проводити щосекунди, а лише при зміні даних. Це можна реалізувати за допомогою *BroadcastReceiver*. *BroadcastReceiver* – це компонент для отримання зовнішніх подій і реакції на них, це об'єкт, який починає виконувати дії при отриманні якого-небудь сигналу (*Intent*). У даного класу тільки один метод *onReceive*. У даному випадку сигналом є зміна параметрів батареї (*ACTION_BATTERY_CHANGED*).

```
BroadcastReceiver receiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        battV.setText("");
        battV.append("Максимальне значення: " +
        intent.getIntExtra(BatteryManager.EXTRA_SCALE, -1) + "%");
        battV.append("\nПоточне значення: " +
        intent.getIntExtra(BatteryManager.EXTRA_LEVEL, -1) + "%");
        battV.append("\nТемпература: " + (float)
        intent.getIntExtra(BatteryManager.EXTRA_TEMPERATURE, -1) / 10.0f + "°C");
        battV.append("\nНапруга: " + (float)
        intent.getIntExtra(BatteryManager.EXTRA_VOLTAGE, -1) / 1000.0f + "V");
        battV.append("\nТехнологія: " +
        intent.getStringExtra(BatteryManager.EXTRA_TECHNOLOGY));
        battV.append("\nСтан: " +
        Func.batteryHealth(intent.getIntExtra(BatteryManager.EXTRA_HEALTH, -1)));
        battV.append("\nСтатус: " +
        Func.batteryStatus(intent.getIntExtra(BatteryManager.EXTRA_STATUS, -1)));
    }
}
```

```
battV.append("\nДжерело: " +  
Func.batteryPlugged(intent.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1)));  
}  
};  
registerReceiver(receiver, new IntentFilter(Intent.ACTION_BATTERY_CHANGED));
```

Інформацію про мобільну мережу можна отримати за допомогою даного об'єкту, використовуючи методи класу *TelephonyManager*:

```
TelephonyManager  
tm=(TelephonyManager) getSystemService(Context.TELEPHONY_SERVICE);
```

Інформацію про дисплей можна отримати за допомогою даного об'єкту, використовуючи методи класу *DisplayMetrics*:

```
DisplayMetrics metrics = new DisplayMetrics();  
getWindowManager().getDefaultDisplay().getMetrics(metrics);
```

Тест на наявність битих пікселів передбачає зображення кольорів (червоний, зелений, синій, чорний білий) у повний екран. Користувач повинен самостійно оглянути екран і пересвідчитись чи всі пікселі одного кольору. Зміна кольорів відбувається при натиску на екран, при натиску на останній колір буде подано вібро сигнал, користувач побачить повідомлення «Тест завершено» «Для виходу натисніть ще раз».

Тест на вібрацію, вмикає вібрацію на 5 секунд, після чого зображується повідомлення «Тест завершено».

Висновки. IDE Android Studio зручне та просте середовище для створення мобільних додатків. Що спонукає розробників розробляти та розвивати додатки на базі операційної системи Android, що в свою чергу популяризує її і приваблює користувачів. Тому з швидким розвитком ПЗ та смартфонів, а також слідуючи вимогам часу, створення додатків для тестування смартфона є доволі актуальним. А відкритість операційної системи дозволяє отримувати системні дані.

1. Broadcast (Широковещательные сообщения) <http://developer.alexanderklimov.ru/android/theory/broadcast.php>
2. Блэйк Мик Программирование под Android. – СПб.: Санкт-Петербург, 2012. – 496 с.
3. Здолбіцька Н.В., Здолбіцький А.П., Найдюк Ю.Ю. Програмне забезпечення для керування системою сенсорів на базі ОС Android // Тези Всеукраїнської науково-практичної інтернет-конференції «Сучасні методи, інформаційне та програмне забезпечення систем управління організаційно-технологічними комплексами», 28 квітня 2015 р. – Луцьк, 2015. – С. 41-42.
4. Лучшие операционные системы для смартфонов <https://keddr.com/2017/04/luchshie-operatsionnyie-sistemyi-dlya-smartfonov/>
5. Разработка, Руководства по API, Операции <https://developer.android.com/guide/components/activities.html?hl=ru>
6. Сенсори. Прискорення, орієнтація [Електронний ресурс]. – Режим доступу: <http://startandroid.ru/ru/uroki/vse-uroki-spiskom/287-urok-137-sensory-uskorenie-orientatsija.html>.
7. Шолом П.С., Здолбіцький А.П., Жигаревич О.К., Яручик В.Л. Роботизована система з дистанційним керуванням // Міжвузівський збірник "Комп'ютерно-інтегровані технології: освіта, наука, виробництво" - 2015. - № 19. - С. 86-90.

УДК 004.912

Коломоєць Г.П., к. ф.-м. н.

Запорізька державна інженерна академія

ПРОГРАМА ПІДГОТОВКИ ДО ДРУКУ ДИПЛОМІВ ПРО ВИЩУ ОСВІТУ ЗА ДАНИМИ ЄДЕБО ІЗ ВИКОРИСТАННЯМ СУЧАСНИХ ТЕХНОЛОГІЙ JAVA

Коломоєць Г.П. Програма підготовки до друку дипломів про вищу освіту за даними ЄДЕБО із використанням сучасних технологій Java. Вирішено актуальну задачу автоматизації підготовки до друку дипломів про вищу освіту за даними здобувачів вищої освіти, експортованими з Єдиної Державної Електронної Бази з питань Освіти (ЄДЕБО). Мета роботи: виконати аналіз структури та форматів даних, які можуть бути експортовані модулем "Замовлення документів" ЄДЕБО, та знайти ефективні засоби платформи Java для розробки програмного додатку для генерації PDF-документів, що представляють собою дипломи про вищу освіту груп здобувачів вищої освіти. За даними групи здобувачів вищої освіти, експортованими з ЄДЕБО в форматі eXtensible Markup Language (XML), за допомогою засобів Java-бібліотеки iText 7 виконується генерація документа у форматі PDF, що містить сторінки дипломів з даними здобувачів із фоновими зображеннями та інформацією, які залежать від рівня вищої освіти та інших параметрів навчання здобувача. Використовувались технологія Java Architecture for XML Binding для відображення експортованих даних у відповідні об'єкти користувацьких класів, високорівневі та низькорівневі засоби бібліотеки iText версії 7.1.0 для побудови PDF-документу, засоби інтернаціоналізації Java для відображення інформації на українській та англійській мовах. Аналіз структури XML-файлу з даними здобувачів вищої освіти, експортованими з ЄДЕБО, виявив неточності, зокрема, пов'язані з невірно вибраними назвами для тегів, що зберігають дати закінчення навчального закладу та дати видачі документа. В роботі запропонований спосіб додавання фонового рисунку до сторінок багатосторінкового PDF-документа без відповідного збільшення розміру файлу. Апробація влітку 2017 р при випуску здобувачів Запорізької державної інженерної академії рівнів вищої освіти "бакалавр" та "спеціаліст" та взимку 2018 р - при випуску здобувачів рівня вищої освіти "магістр" довела ефективність розробленого програмного забезпечення.

Ключові слова: ЄДЕБО, диплом про вищу освіту, XML, JAVA, PDF, iText.

Коломоєць Г.П. Программа подготовки к печати дипломов о высшем образовании по данным ЕГЭБО с использованием современных технологий Java. Решена актуальная задача автоматизации подготовки к печати дипломов о высшем образовании по данным соискателей высшего образования, экспортируемым из Единой Государственной Электронной Базы по вопросам Образования (ЕГЭБО). Цель работы: выполнить анализ структуры и форматов данных, которые могут быть экспортированы модулем "Заказ документов" ЕГЭБО, и найти эффективные средства платформы Java для разработки программного приложения для генерации PDF-документов, представляющих собой дипломы о высшем образовании групп соискателей высшего образования. По данным группы соискателей высшего образования, экспортируемым из ЕГЭБО в формате eXtensible Markup Language (XML), с помощью средств Java-библиотеки iText 7 выполняется генерация документа в формате PDF, содержащего страницы дипломов данным соискателей с фоновыми изображениями и информацией, которые зависят от уровня высшего образования и других параметров обучения соискателя. Использовались технология Java Architecture for XML Binding для отображения экспортируемых данных в соответствующие объекты пользовательских классов, высокоуровневые и низкоуровневые средства библиотеки iText версии 7.1.0 для построения PDF-документа, средства интернационализации Java для отображения информации на украинском и английском языках. Анализ структуры XML-файла с данными соискателей высшего образования, экспортируемыми из ЕГЭБО, обнаружил неточности, в частности, связанные с неверно выбранными названиями для тегов, сохраняющих даты окончания учебного заведения и даты выдачи документа. В работе предложен способ добавления фонового рисунка к страницам многостраничного PDF-документа без соответствующего увеличения размера файла. Апробация летом 2017 г при выпуске соискателей Запорожской государственной инженерной академии уровней высшего образования "бакалавр" и "специалист" и зимой 2018 г - при выпуске соискателей уровня высшего образования "магистр" доказала эффективность разработанного программного обеспечения.

Ключевые слова: ЕГЭБО, диплом о высшем образовании, XML, JAVA, PDF, iText.

Kolomoiets G.P. Software of preparation for the printing of diplomas about higher education by data of the USEDEI with the use of modern Java technologies. The actual task of automating of preparation for the printing of diplomas about higher education according to the data of higher education seekers exported from the Unified State Electronic Database on Education Issues (USEDEI) was solved. The purpose of the work is to analyze the structure and data formats that can be exported by the "Ordering Documents" module of USEDEI and to develop software for generating PDF documents, which are diplomas of higher education of groups of higher education seekers. By data exported from the USEDEI in the eXtensible Markup Language (XML) format of higher education seekers group, using the iText 7 Java library tools, a PDF document is generated that contains the diploma pages to seekers with background images and information that depend on the higher education level and other parameters of seeker education. We used the Java Architecture for XML Binding technology to mapping the exported data to the corresponding objects of the user classes, we used high-level and low-level tools of the iText library version 7.1.0 for the construction of the PDF document, Java internationalization tools for to display information in Ukrainian and English. Analysis of the structure of the XML file with the data of higher education seekers exported from the USEDEI found inaccuracies, in particular, related to incorrectly selected names for tags that contain the graduation dates and the issue date of the document. In this paper, a method is proposed for adding a background image to pages of a multipage PDF document without a corresponding increase in the file size. The effectiveness of the developed software was proved by approbation in the summer of 2017 for bachelors and specialists of Zaporizhzhia State Engineering Academy and in the winter of 2018 - for masters.

Keywords: USEDEI, diploma of higher education, XML, JAVA, PDF, iText.

Постановка проблеми. Відповідно до статті 7 Закону України "Про вищу освіту" [1], заклади вищої освіти отримали право на виготовлення документів про вищу освіту: дипломів та додатків до них, які повинні містити перелік інформації, затверджений наказом Міністерства освіти і науки України від 12 травня 2015 р. № 525 "Про затвердження форм документів про вищу освіту (наукові ступені) державного зразка та додатків до них, зразка академічної довідки" [2]. Оскільки вся інформація, яка відтворюється в дипломах про вищу освіту, відповідно до статті 8 Закону України "Про вищу освіту" знаходиться в Єдиній державній Базі з питань Освіти (ЄДЕБО) [1], можлива автоматизація підготовки дипломів при вищій освіті для друку із використанням можливостей експорту даних з ЄДЕБО.

І хоча послуги по виготовленню дипломів можуть виконувати інші організації [3], на наш погляд, для закладу вищої освіти актуальною є розробка власного програмного забезпечення для можливості оперативного друку документів та внесення змін до функціональності такого програмного забезпечення. Окрім того, представляє певний інтерес дослідження і використання сучасних засобів програмної інженерії, зокрема такої розвиненої платформи з вільно розповсюджуваним програмним кодом, як Java, для ефективної розробки програмного забезпечення.

Для вирішення проблеми було необхідно:

- виконати аналіз структури даних та порівняння форматів XML-файлів експорту даних документів про освіту з ЄДЕБО [4-5], обравши дані, актуальні для програмного забезпечення, що розробляється;
- виконати проектування документу у форматі PDF із інформацією дипломів про вищу освіту здобувачів вищої освіти відповідно до наказу МОНУ від 12 травня 2015 р. №525 [2];
- виконати проектування архітектури програмного забезпечення, провести конструювання програмного забезпечення та підготовку його до дистрибуції;
- виконати апробацію роботи програмного забезпечення для вихідних даних здобувачів різних ступенів вищої освіти, експортованих з ЄДЕБО.

Аналіз досліджень. Традиційно парсинг XML в Java виконується з використанням таких інтерфейсів прикладного програмування (API) Java, як простий API для XML (Simple API for XML - SAX) або об'єктна модель документа (Document Object Model - DOM). Засоби першого відрізняються послідовним зчитуванням розмітки XML-файла та передачею програмному додатку подій при зустрічі розпізнаних елементів XML, які зазвичай оброблюються за допомогою функцій зворотного виклику. SAX-парсери відрізняються гарною продуктивністю, але не дозволяють змінювати структуру XML-файлу. DOM API базується на попередньому аналізі всього XML-файла та створенні для кожного його тега об'єкту-вузла *Node*, який інкапсулює ім'я, атрибути та значення атрибутів тега. Цей API дозволяє не тільки зчитувати у довільному порядку, але й змінювати XML-документ [6].

Але обидва класичних підходи до парсинга XML вимагають досить великого кодування для отримання значень тегів та їх атрибутів, які потім, як правило, інкапсулюються в об'єкти користувацьких класів, що надалі використовуються у програмі. Наступний рівень абстрагування - безпосереднє зчитування даних XML-файла та збереження їх у об'єктах користувацьких класів забезпечує технологія Java Architecture for XML Binding (JAXB) [7]. Для кожного XML-документа за допомогою мови описання структури XML-документа XML Schema може бути побудована відповідна схема. У разі попередньої підготовки користувацького класу, що інкапсулює необхідні у програмі властивості, які заповнюватимуться значеннями тегів та атрибутів XML-документу, така схема може бути виведена засобами JAXB автоматично за допомогою відповідних анотацій властивостей класів. Таке анотування забезпечує процедуру відображення (bind) XML-схеми на користувацький клас (рис. 1). Після такого відображення можливе використання технології демаршалінга (unmarshaling), яка дозволяє отримувати об'єкти користувацьких класів з властивостями зі значеннями із відповідних тегів та атрибутів XML-документу. Також можлива зворотна генерація XML-документу за даними об'єктів - технологія маршалінга (marshaling) [7].

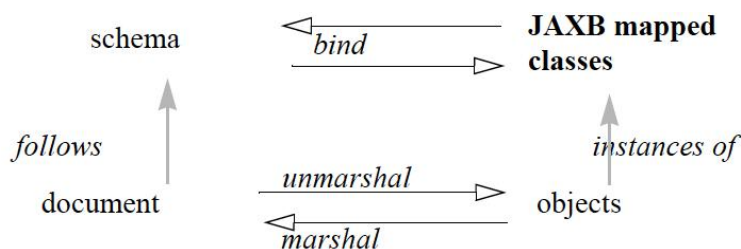


Рис. 1 - Відображення XML-файлу на об'єкти [7]

В роботі у якості формату вихідного документу з дипломами здобувачів вищої освіти обраний формат PDF, який дозволяє відображати та друкувати незмінний документ із даними різного типу на різних платформах та пристроях. Оскільки вбудованих до API платформи Java засобів генерації PDF-документів не існує, потрібно було обрати сторонню бібліотеку для забезпечення цього функціоналу. Дослідження в Інтернет показали, що найбільшою популярністю користуються бібліотеки Apache PDF та iText, а серед них користувачі більше застосовують iText [8]. Перевагами цієї бібліотеки є добра документованість, наявність книг [9-11] та сайту підтримки з великою кількістю прикладів використання засобів бібліотеки, розвинений функціонал, що дозволяє задовольнити усі потреби програмістів при роботі з PDF-файлами, на час написання статті актуальною була версія 7.1.0 бібліотеки.

Бібліотека iText надає як високорівневі засоби роботи з PDF-документом за допомогою об'єктів класів Document, які можуть містити об'єкти класів Paragraph, List, Image та інших, так і низькорівневі засоби - об'єкти класів PdfWriter, PdfDocument та PdfCanvas, які дозволяють, наприклад, програмно виводити графіки та додавати блоки тексту у довільні області сторінки.

Виклад основного матеріалу і обґрунтування отриманих результатів. Аналіз структури XML-файлу, експортованого з ЄДЕБО [4] та порівняння його зі структурою аналогічного файлу, експортованого попередньою версією програмного забезпечення ЄДЕБО [5] дозволяє заключити:

1) розробники змінили підхід до структури XML-файлу, а саме, якщо раніше дані для кожного здобувача зберігались як атрибути тегу <documentdata>, то зараз вони зберігаються у окремих тегах, окрім того, використовується CamelCase-нотація, що спрощує сприйняття назв XML-елементів (рис. 2);

2) новий формат містить більше даних, ніж попередній (74 проти 46), але відрізняється, на наш погляд, надлишковістю: для деяких даних зберігається і ідентифікатор, і назва, наприклад, "Ідентифікатор відзнаки" <AwardTypeId> і "Назва відзнаки" <AwardTypeName> (0 - без відзнаки, 3 - з відзнакою – незрозуміло, чому не 1). Теж саме відноситься до ідентифікатора і назви джерела даних, ідентифікатора і назви типу вихідного документа для дубліката, ідентифікатора і назви типу фінансування, ідентифікатора і назви форми навчання, ідентифікатора і назви курсу, ідентифікатора і назви статі. Однак явно надлишкові елементи - прізвище, ім'я та по-батькові випускника разом (fio та fioen) видалені.

3) на наш погляд, присутня помилка у виборі назви для дати закінчення навчального закладу <IssueDate> та дати видачі документа <GraduateDate> - з точки зору англійської мови потрібно було б назвати ці теги навпаки. Як результат, для випадку дублікату диплому, коли не співпадають зазначені дати, ми отримуємо:

```
<GraduateDate>16.10.2017 0:00:00</GraduateDate>
<IssueDate>30.06.1988 0:00:00</IssueDate>
```

що без документації може сприйматись як дата закінчення пізніша ніж дата видачі диплому.

4) перевагами нового формату є наявність додаткових тегів, що дозволяють зберігати назву освітньої програми здобувача двома мовами, а також теги для збереження даних назв та розрядів професій;

5) фактичний формат дати та часу тегів <GraduateDate> та <IssueDate> не відповідає документації (там вказаний формат без часу: ДД.ММ.РРРР [4]).

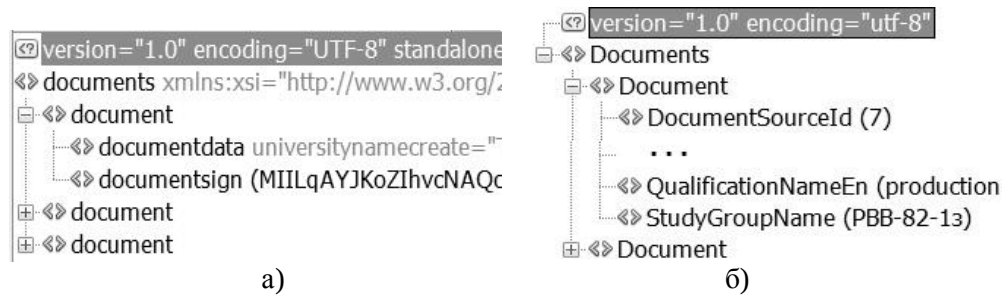


Рис. 2 - Структура XML-файлу з даними здобувачів вищої освіти, експортованими з ЄДЕБО: а) у попередній версії програмного забезпечення ЄДЕБО, б) у поточній версії програмного забезпечення ЄДЕБО

Для підготовки до друку така велика кількість даних була надлишковою, тому у спроектованому на архітектурному рівні моделі класі `Diploma` було обрано лише 22 елементи даних, які відображені у структурі класу на рис. 3. Відображення XML-документу на об'єкти цього класу конфігурується JAXB-анотаціями [7].

Для організації демаршалінга довільних класів архітектурного рівня моделі з використанням технології узагальнень (Java Generics) був розроблений універсальний парсер з методом демаршалінга XML-файлу у об'єкти аргументу узагальненого типу, вихідний код класу якого наведений нижче:

```
public class JAXBParser<T> {

    final Class<T> typeParameterClass;

    public JAXBParser(Class<T> typeParameterClass) {
        this.typeParameterClass = typeParameterClass;
    }

    public T getObject(File file) throws JAXBException {

        JAXBContext context =
            JAXBContext.newInstance(typeParameterClass);
        Unmarshaller unmarshaller =
            context.createUnmarshaller();
        T object = (T) unmarshaller.unmarshal(file);
        return object;
    }
}
```

Такий клас розміщений на рівні сервісів (рис. 4). Об'єкт цього класу ініціалізується об'єктом `Class` для кореневого класу моделі XML-документу, у нашому випадку:

```
JAXBParser<Diplomas> parser = new JAXBParser<>(Diplomas.class);
```

Виклик метода `getObject(File file)` цього об'єкта з передачею як аргументу об'єкта `File`, пов'язаного з XML-файлом з даними, повертає об'єкт `Diplomas` зі списком об'єктів `Diploma` з даними здобувачів вищої освіти.

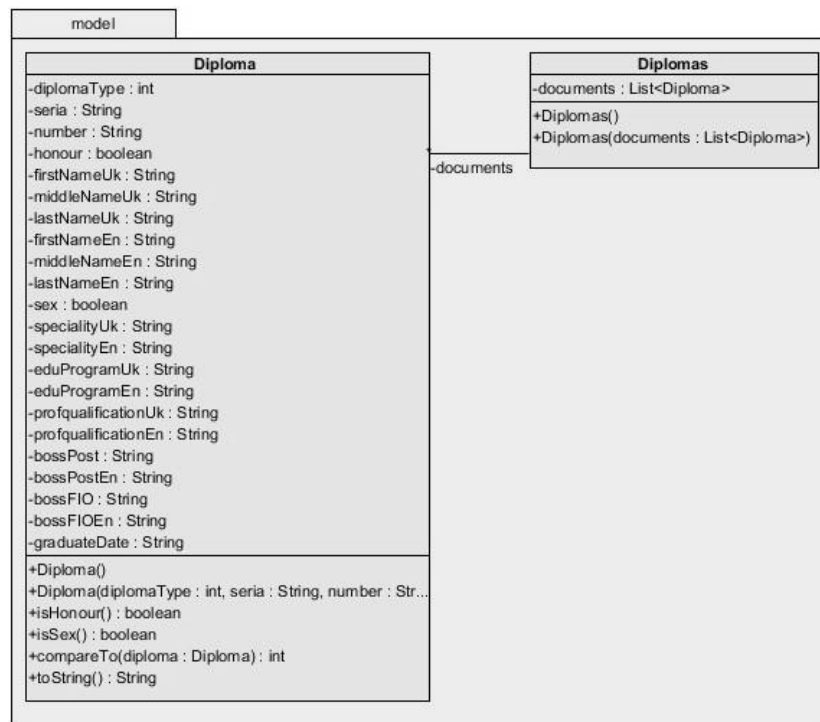


Рис. 3 - Класи архітектурного рівня моделі

Після отримання об'єктів з даними дипломів здобувачів вищої освіти настає етап генерування PDF-документу із дипломами, готовими до друку, за допомогою засобів бібліотеки iText. Перелік jar-файлів з засобами бібліотеки iText, який вказується у [10], для реалізації програмного забезпечення, що описується, був надлишковим. Дослідженням встановлена необхідність тільки таких файлів бібліотеки: itext7-kernel-7.0.2.jar, itext7-io-7.0.2.jar, itext7-layout-7.0.2.jar. Також знадобилися файли додаткових бібліотек slf4j-api-1.7.25.jar та slf4j-simple-1.7.25.jar - для системи організації журналів (логів) та commons-io-2.5.jar - для завантаження байтів фонових зображень при запуску програми із jar-файлу.

На рис.5 наведений приклад сторінки згенерованого PDF-документа із зазначенням методів класу PdfGenerator, відповідальних за побудову частин сторінки.

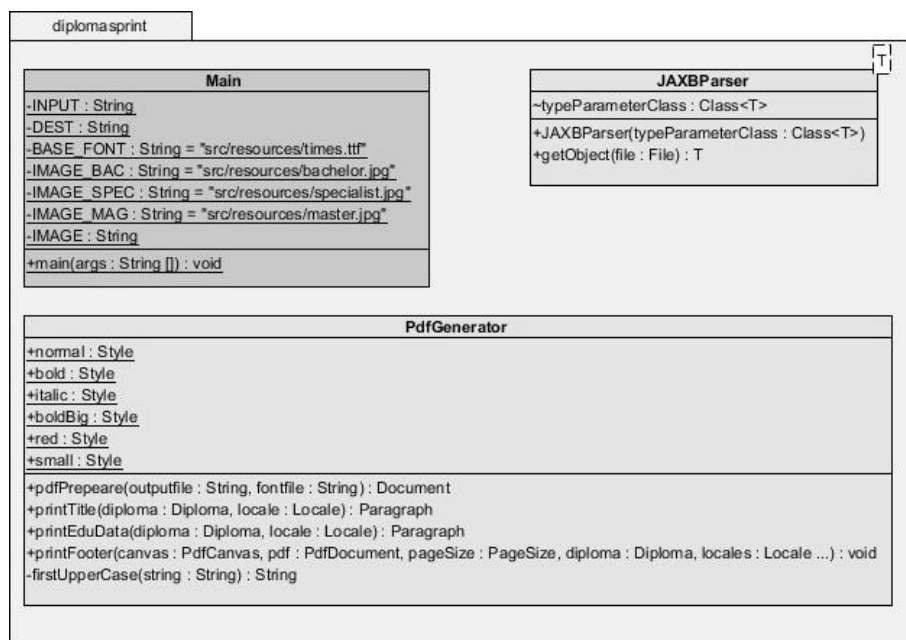


Рис. 4 - Класи архітектурного рівня сервісів



Рис. 5 – Зображення прикладу сторінки згенерованого PDF-документа із зазначенням методів PdfGenerator, відповідальних за побудову частин сторінки

Фоновий рисунок відрізняється кольором захисних елементів для дипломів різних рівнів вищої освіти: бакалавр, спеціаліст та магістр, та автоматично підставляється у залежності від значення властивості `diplomaType` об'єкту `Diploma`. Звертаємо увагу на використання об'єктів утилітарного класу `Java Locale` для організації двомовного виведення інформації. Метод `Document pdfPrepeare(String outputfile, String fontfile)` класу `PdfGenerator` за допомогою засобів бібліотеки `iText` виконує підготовку до виведення інформації у PDF-документ [11]:

- визначає розмір, орієнтацію та відступи від краю сторінки диплому;
- створює об'єкт `PdfFont` з підтримкою символів кирилиці;
- визначає об'єкти стилів шрифту різного розміру та написання, також передбачений стиль шрифту червоного кольору для виведення назви диплому з відзнакою;
- конфігурує виведення інформації у дві колонки та визначає розміри колонок.

Цей метод приймає посилання на місце розташування результуючого PDF-файлу та на файл шрифту (можливо використовувати звичайний True Type шрифт з символами кирилиці). Повертає метод об'єкт високорівневого класу `Document`, який і є результуючим PDF-документом.

Метод `Paragraph printTitle(Diploma diploma, Locale locale)` виводить інформацію про рівень вищої освіти диплому, признак відзнаки, якщо є (у цьому випадку назва диплому виводиться червоним кольором та додається рядок про відзнаку), серію та номер диплому, прізвище, ім'я та по-батькові здобувача та назву навчального закладу, який він закінчив і рік завершення навчання. Особливістю методу є використання засобів інтернаціоналізації `Java`, включаючи використання пакетів ресурсів (`ResourceBundle`) разом з файлами властивостей та форматування повідомлень за допомогою засобів класу `MessageFormat`. Ці засоби дозволили повторно використовувати код методу та адаптувати вивід інформації в залежності від об'єктів `Locale`, які передаються методу як аргумент. Також за допомогою вказаних технологій коректуються закінчення україномовних слів у залежності від статі здобувача.

Метод `Paragraph printEduData(Diploma diploma, Locale locale)` виводить інформацію про параметри здобутої випускником кваліфікації: рівень вищої освіти, спеціальність (або напрям для бакалавра), назву освітньої програми та професійну кваліфікацію (за наявності). Використовуються ті ж самі засоби, як і в методі `printTitle`.

Додання нижньої частини диплому з посадою, прізвищем та підписом керівника та іншими даними на високому рівні виявилось проблематичним, оскільки внаслідок різного розміру даних, що виводяться описаними вище методами, його положення б змінювалось. Тому було прийнято

рішення в методі `void printFooter(PdfCanvas canvas, PdfDocument pdf, PageSize pageSize, Diploma diploma, Locale... locales)` використати об'єкти низькорівневого класу `PdfCanvas` та класу `Canvas`, що зв'язує низькорівневі та високорівневі засоби бібліотеки `iText` і дозволяє абсолютне позиціонування інформації, що виводиться [11]. Окрім низькорівневих об'єктів `PdfCanvas` та `PdfDocument`, метод приймає об'єкт розміру сторінки, документ з даними добувача та масив локалей, що використовуються (передаються як аргументи змінної довжини `varargs`). Оскільки тег дати видачі диплома `<GraduateDate>` зберігає окрім дати і час, і потрібно виводити дату у повному форматі з урахуванням правил виведення для української та англійської локалі, в методі були застосовані засоби `Java Date and Time API` – клас `DateFormatter`.

Клас `Main` містить константи, що визначають шляхи до файлів-ресурсів: файлу шрифту та файлів фонових зображень для трьох рівнів вищої освіти. Назва вихідного XML-файлу з даними користувачів передається при запуску програми як аргумент командного рядку, а результуючий PDF-файл з назвою, як у XML-файла, але з розширенням `.pdf`, зберігається у підкаталозі `result` каталогу, з якого виконувався запуск програми. Головний метод додатку `void main(String[] args)`, розміщений в цьому класі, за допомогою об'єкту класу `JAXBParser`, параметризованого класом архітектурного рівня моделі `Diplomas`, отримує список об'єктів `Diploma`, після чого список відсортовується за прізвищами здобувачів.

Цікавою особливістю програми є спосіб додавання фонових рисунків до сторінок PDF-документу. Річ у тому, що багатократне додавання звичайного рисунку – об'єкту `Image` до документу не збільшує розмір PDF-файлу, тобто байти рисунку у документі зберігаються тільки один раз, а дублюються посилання на них [11]. Але для додавання фонових рисунків доводиться використовувати низькорівневий об'єкт `PdfCanvas`, специфічний для кожної сторінки PDF-документу. При цьому з'являється проблема необґрунтованого збільшення розміру результуючого PDF-файлу пропорційно кількості сторінок з однаковим фоновим зображенням в ньому [12]. Вказану проблему вдалося вирішити додаванням до `PdfCanvas` об'єкту `PdfImageXObject`, що є "обгорткою" над фоновим рисунком, в обох випадках об'єкти зображень створюються через об'єкт `ImageData`:

```
ImageData imgData = ImageDataFactory.create(IMAGE);
```

Але у першому випадку об'єкт `ImageData` додається безпосередньо:

```
canvas.addImage(imgData, pageSize, false);
```

А у разі оптимізації – через об'єкт `PdfImageXObject`:

```
PdfImageXObject imgObject = new PdfImageXObject(imgData);  
canvas.addXObject(imgObject, rectangle);
```

Аргумент `rectangle` визначає область сторінки для додавання зображення, при розтягненні зображення на всю сторінку він може визначатись через розмір сторінки:

```
Rectangle rectangle = new Rectangle(pageSize.getWidth(),  
                                     pageSize.getHeight());
```

Дослідженням результуючих PDF-файлів у програмі `PDFXplorer` [13] було з'ясовано, що безпосереднє додавання фонових зображень до об'єкту `PdfCanvas` призводить до наявності декількох об'єктів `PDF XObject`. Це зумовлено тим, що метод `PdfXObject addImage(ImageData image, Rectangle rect, boolean asInline)` класу `PdfCanvas` створює кожен раз окремий об'єкт `PdfXObject`, навіть для одного і того ж об'єкту `ImageData`. Натомість, використання методу `PdfCanvas addXObject(PdfXObject xObject, Rectangle rect)` того ж класу у якості аргументу приймає об'єкт `PdfXObject`, який відповідно до специфікації `PDF` – є зовнішнім до основного контенту об'єктом [14]. Вочевидь, тому останній метод використовує посилання на один і той же об'єкт, який "обгортає" фоновий рисунок. Було виконано порівняння розміру згенерованих представленим програмним забезпеченням багатосторінкових файлів з однаковими фоновими рисунками, що були додані до канви безпосередньо та з використанням `PdfImageXObject`. Для файлу формату `JPG` розміром 1981250 байт 5-сторінковий `PDF`-документ з

безпосереднім доданням фонового рисунку до канви мав розмір 9909084 байта, у той час як 5-сторінковий PDF-документ з доданням того ж фонового рисунку до канви через об'єкт `PdfImageXObject` мав розмір 1983364 байти, тобто у 4,996 разів менше.

Після додання фонового зображення у методі `void main(String[] args)` до документу додаються об'єкти параграфів, що створюються вище описаними методами об'єкту `PdfGenerator`. Ці операції виконуються у циклі з кількістю ітерацій, що дорівнює кількості дипломів здобувачів вищої освіти у вихідному XML-файлі. При цьому кінець першої колонки організовується додаванням до PDF-документу об'єкта `AreaBreak`, а кінець сторінки – об'єкта `AreaBreak` з передачею його конструктору значення `AreaBreakType.NEXT_PAGE` як аргументу.

Оскільки взаємодія програми з користувачем дуже проста: запуск програми з командного рядка з зазначенням імені XML-файлу як параметра, було вирішено підготувати дистрибутив програми у вигляді jar-файлу. При цьому було вирішено за доцільне упакувати разом з файлами програми файли ресурсів: файл шрифту, файли фонових зображень, та файли властивостей для української та англійської локалей. Завантаження файлів шрифту та зображень було організоване за допомогою методу `static byte[] toByteArray(InputStream input)` класу `IOUtils` бібліотеки Apache Commons IO [15]:

```
byte[] fontBytes = IOUtils.toByteArray(  
    Main.class.getResource(fontfile).openStream());
```

та створенням з масиву байтів об'єкту `PdfFont` (та об'єктів `ImageData`):

```
PdfFont font = PdfFontFactory.createFont(fontBytes,  
    PdfEncodings.IDENTITY_H);
```

або

```
ImageData imgData = ImageDataFactory.create(imageBytes);
```

Для запуску програми на комп'ютері користувача повинно бути встановлено середовище для запуску програм Java (JRE) або комплект розробника програм на Java (JDK) версії не нижче 7. При цьому команда запуску програми буде

```
java -jar DiplomasPrint.jar MET-13-1.xml
```

У разі встановлення на комп'ютері користувача останньої 9 версії JRE або JDK команда запуску додатку зміниться внаслідок переходу Java до концепції модулів та виключення зі стандартного модуля `java.se` засобів Java EE, у тому числі JAXB API [16]. У цьому випадку потрібно підключати модуль з засобами JAXB:

```
java -jar --add-modules java.xml.bind DiplomasPrint.jar MET-13-1.xml
```

Представлене програмне забезпечення було апробоване при підготовці до друку дипломів здобувачів вищої освіти у Запорізькій державній інженерній академії: бакалаврів та спеціалістів у червні 2017 р та дипломів магістрів у січні 2018 р. Апробація довела ефективність програми за рахунок зменшення часу на друк та зменшення помилкових роздруківок, оскільки раніше виконувалася окремих друк фонового рисунка та інформації диплому.

Висновки та перспективи подальшого дослідження. Аналіз структури XML-файлу з даними здобувачів вищої освіти, експортованими з ЄДЕБО, виявив неточності, зокрема, пов'язані з невірно вибраними назвами для тегів, що зберігають дати закінчення навчального закладу та дати видачі документа.

Використання технології відображення елементів XML-файлів на об'єкти користувацьких класів (JAXB) суттєво зменшили обсяг кодування для реалізації парсингу та підвищили надійність роботи програми. Розроблений типізований JAXB парсер обмежує використання типів аргументом типу, що надає відомі переваги [17]. Застосування засобів технології інтернаціоналізації додатків Java дозволило повторно використовувати методи генерування даних дипломів на українській та англійській мовах. Використання технології форматування повідомлень дозволило забезпечити необхідну зміну порядку слів та корекцією закінчень для різних локалей.

В роботі запропонований спосіб додання фонового рисунку до сторінок багатосторінкового PDF-документу без відповідного збільшення розміру файлу.

Виконана адаптація програмного коду для формування JAR-дистрибутиву програмного забезпечення та досліджені особливості його запуску під найновішою 9 версією Java.

Результати аналізу структури XML-файлу, з даними експортованими з ЄДЕБО, надіслані до розробників програмного забезпечення ЄДЕБО зі сподіванням їх врахування при модернізації ЄДЕБО.

1. "Про вищу освіту" Закон України від 1 липня 2014 р № 1556-VII // Відомості Верховної Ради (ВВР). – К.: ПАРЛАМЕНТСЬКЕ ВИДАВНИЦТВО, 2014, № 37-38, – С. 2004.
2. "Про затвердження форм документів про вищу освіту (наукові ступені) державного зразка та додатків до них, зразка академічної довідки" наказ Міністерства освіти і науки України від 12 травня 2015 р. № 525 [Електронний ресурс] / Сайт "Законодавство України". – Режим доступу: <http://zakon3.rada.gov.ua/laws/show/z0551-15>.
3. Виготовлення документів про вищу освіту [Електронний ресурс] / Сайт інформаційно-виробничої системи "Освіта". – Режим доступу: <https://osvita.net/ua/vigotovlennya-dokumentiv-pro-vishhu-osvitu>.
4. Опис XML-файлу для експорту даних документів про освіту з ЄДЕБО [Електронний ресурс] / Сайт Державного Підприємства "Інфоресурс". – Режим доступу: <https://www.inforesurs.gov.ua/uploads/files/EDEBOFormat-XML-Zamovl-2017.pdf>.
5. Опис XML файлу для експорту даних документів про освіту (з використанням попередньої версії програмного забезпечення ЄДЕБО) [Електронний ресурс] / Сайт Державного Підприємства "Інфоресурс". – Режим доступу: https://www.inforesurs.gov.ua/uploads/files/1434111418_edebo.format-xml-zamovl.dvo.pdf.
6. McLaughlin B. Java and XML. Solutions to Real-World Problems, 3rd Edition / B. McLaughlin, J. Edelson. – Sebastopol : O'Reilly Media, 2009. – 480 p.
7. JSR-000222 Specification. The Java™ Architecture for XML Binding (JAXB) 2.3 Final Release July 11, 2017 [Electronic resource] / – Access mode: http://download.oracle.com/otndocs/jcp/jaxb-2_3-mrel3-eval-spec/index.html.
8. Compare Apache PDFBox and iText's popularity and activity [Electronic resource] – Site Awesome Java / – Access mode: <https://java.libhunt.com/project/apache-pdfbox/vs/itext>.
9. Lowagie B. iText in Action. Second Edition. / B. Lowagie. – Stamford : Manning Publications Co., 2011. – 618 p.
10. Lowagie B. iText 7: Jump-Start Tutorial. / B. Lowagie. – Lean Publishing, 2016. – 100 p.
11. Lowagie B. iText 7: Building Blocks. / B. Lowagie. – Lean Publishing, 2016. – 224 p.
12. Kolomoets G. iText 7 read background image from jar file [Electronic resource] – Site Stack overflow / – Access mode: <https://stackoverflow.com/questions/44871451/itext-7-read-background-image-from-jar-file>.
13. PDFXplorer [Electronic resource] / Site O2 Solutions – Access mode: <http://www.o2sol.com/pdfexplorer/overview.htm>
14. PDF 32000-1:2008. Document management - Portable Document Format - Part I: PDF 1.7, First Edition [Electronic resource] / Site Adobe Systems Incorporated – Access mode: http://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/PDF32000_2008.pdf.
15. Apache Commons IO [Electronic resource] / Site Apache Commons – Access mode: <https://commons.apache.org/proper/commons-io>.
16. Sander M. Java 9 Modularity. / M. Sander, P. Bakker – Sebastopol : O'Reilly Media, 2017. – 366 p.
17. Naftalin M. Java Generics and Collections / M. Naftalin, Ph. Wadler – Sebastopol : O'Reilly Media, 2007. – 286 p.

УДК 004.5

Костючко С.М., Дацюк С.В., Нагорнюк А.І.

Луцький національний технічний університет

СЕСІЇ ТА АВТОРИЗАЦІЯ КОРИСТУВАЧІВ ІНТЕРНЕТ-МАГАЗИНУ ЗА ДОПОМОГОЮ PYTHON ТА DJANGO

Костючко С.М., Дацюк С.В., Нагорнюк А.І. СЕСІЇ ТА АВТОРИЗАЦІЯ КОРИСТУВАЧІВ ІНТЕРНЕТ-МАГАЗИНУ ЗА ДОПОМОГОЮ PYTHON ТА DJANGO. У даній статті висвітлено аспекти електронної комерції та можливості фреймворка Django для створення інтернет-магазинів. Наведено приклади авторизації та реєстрації користувачів.

Ключові слова: e-commerce, Python, Django, авторизація користувачів.

Костючко С.Н., Дацюк С.В., Нагорнюк А.И. СЕССИИ И АВТОРИЗАЦИЯ ПОЛЬЗОВАТЕЛЕЙ ИНТЕРНЕТ-МАГАЗИНА С ПОМОЩЬЮ PYTHON И DJANGO. В данной статье описано аспекты электронной коммерции и возможности фреймворка Django для создания интернет-магазинов. Приведены примеры авторизации и регистрации пользователей.

Ключевые слова: e-commerce, Python, Django, авторизация пользователей.

Kostiuchko S.M., Datsyuk S.V., Nagornyuk A.I. SESSIONS AND AUTHORIZATION OF INTERNET-STORE USERS USING PYTHON AND DJANGO. This article covers aspects of e-commerce and the capabilities of the Django framework for creating online stores. Examples of authorization and user registration are given.

Keywords: e-commerce, Python, Django, user authorization.

Актуальність. В сучасному інформаційному суспільстві всі користуються інтернет-магазинами. Це зручно, дає змогу порівняти товари і обрати необхідний. Електронна торгівля (*Electronic commerce*) займає вагоме місце у веденні бізнесу: вона забезпечує необхідними товарами і послугами. Завдяки їй, товар може бути доставлений в найвіддаленіші куточки світу. Велике число людей, особливо молодь, дуже часто вважає за краще купувати товари в Інтернеті, ніж в традиційних магазинах. Перш за все, це пояснюється низькими цінами, а також більш широким асортиментом товару.

Вступ. Електронна комерція дозволяє підвищити ефективність внутрішніх операцій компанії, взаємодіяти з клієнтами та постачальниками, не приділяючи при цьому уваги різним обмежуючим факторам, таким, як географічне положення. Діловими сторонами в електронній комерції можуть бути будь-які організації і компанії, що мають доступ до Інтернет. За допомогою електронної комерції компанії можуть швидко знайти собі відповідних постачальників, клієнтів, ділових партнерів і організовувати свій бізнес без постійних поїздок на переговори, укладання договорів і т.д. За допомогою електронної комерції можна здійснювати безліч бізнес-операцій: пошук замовників і постачальників, поширення інформації про товари і послуги, оплата (за допомогою електронних платежів), доставка і розповсюдження електронних товарів. За участю в бізнес процесі, електронну комерцію можна розділити на «бізнес для бізнесу», «бізнес для клієнта» і «бізнес для адміністрації». Розглядаючи змістовні характеристики електронної комерції, слід зазначити властиві їй широкий спектр бізнес-операцій і угод, а також комунікаційних технологій. Перелік бізнес-операцій і угод включає встановлення контакту, обмін інформацією між постачальником і замовником, оплату товару або послуги з використанням електронного переказу грошей, перед- і післяпродажну підтримку реалізованого електронного продукту (інструкції по використанню, відповіді на питання), поширення і безпосередню доставку продуктів, які можуть поширюватися електронним шляхом.

Історія розвитку. Історія електронної комерції бере свій початок у 1990 році, коли Тім Бернерс-Лі створив перший веб-сервер і браузер. Він був відкритий для комерційного використання в 1991 році. У 1994 році відбулися інші відкриття, наприклад, онлайн-банкінг та відкриття інтернет-магазину піци «Pizza Hut». У тому ж році Netscape представила SSL-

шифрування даних, переданих в мережі, яке стало необхідним для безпеки інтернет-магазинів. Крім того, в 1994 році німецька компанія Intershop представила свою першу систему інтернет-магазинів. У 1995 році Amazon запустила свій інтернет-магазин, а в 1996 році з'явився eBay. Станом на 2017 рік Amazon має окремі сайти для роздрібною торгівлі у США, Великій Британії, Ірландії, Франції, Канаді, Німеччині, Італії, Іспанії, Нідерландах, Австралії, Бразилії, Японії, Китаї, Індії та Мексиці. Чистий прибуток Amazon складає \$2,371 млрд. (2016).

Переваги електронної комерції. У багатьох випадках електронна комерція дозволяє скоротити шлях перепродажу продукту від виробника до споживача. Це можливо завдяки використанню Інтернет-технологій, що надають можливість ефективної прямої взаємодії з кінцевим споживачем, тому компанії можуть виконувати роль, яку традиційно виконували проміжні постачальники. Це також дозволяє накопичувати інформацію про усі продажі та про усіх клієнтів, що у свою чергу дозволяє виконати досконалий бізнес-аналіз та маркетингові дослідження. Це є великою перевагою у конкурентній боротьбі. Електронне середовище широко використовується для доставки цифрового медіа-контенту (музика, фільми, преса тощо), корисної інформації, освітніх матеріалів, а також компаніями-виробниками програмного забезпечення для його продажу. Найбільшою перевагою електронної комерції є суттєве зниження витрат на оформлення угоди та її подальше обслуговування. Тому бізнес-процеси, які можуть бути переведені на електронну основу мають потенціал зниження витрат на них, що у свою чергу призводить до зниження собівартості товару чи послуги. Найвідомішим прикладом здійснення електронної комерції є Інтернет-магазин, який являє собою веб-ресурс з каталогом продукції та можливістю замовлення і оплати товарів, які сподобались покупцю. Все більше компаній у світі впроваджують рішення електронної комерції у своєму бізнесі. Наприклад, всесвітньо відома компанія CISCO не має традиційної мережі дистриб'юторів. Замість того, вона приймає замовлення тільки в електронній формі зокрема зі свого веб-сайту. Інший приклад — це виробники ноутбуків. На своїх веб-сайтах вони розміщують інтерактивні сторінки, де користувач може обрати собі ноутбук за своїми потребами та оформити замовлення і оплату.

Сесії та аутентифікація користувачів в Django. З точки зору організації роботи, будь-який інтернет-магазин можна розділити на дві частини:

Front-end. Це сам сайт інтернет-магазину, поштова та email-розсилка покупцям. Загалом, це все те, з чим стикається покупець.

Back-end. Це робота служби доставки, ціноутворення, постачання, інтеграція різних систем. Загалом, це та частина роботи, яка для покупця залишається «за кадром», але також дуже важлива для безперебійної і надійної роботи.

У даному випадку front-end складається з категорій товарів, особистого кабінету користувача, сторінки авторизації і реєстрації та власне товарів з зображенням (зображеннями), описом, характеристиками, фільтрами для пошуку по атрибутах, оцінками, коментарями споживачів, а також цінами та акційними пропозиціями. Для забезпечення стильового оформлення використовуються каскадні таблиці стилів (CSS3), а також бібліотека (фреймворк) Bootstrap 4. Інтерактивність та перевірка правильності заповнення форм користувачем виконується за допомогою технології JavaScript. В якості back-end використовується мова програмування Python та веб-фреймворк Django, а також реляційна база даних PostgreSQL.

Django містить власну вбудовану систему аутентифікації користувача. Вона обробляє облікові записи користувачів, групи, дозволи, а також сесии користувачів на основі файлів cookie.

Система аутентифікації складається з:

1. Користувачів

2. Дозволів: Двійкових (так/ні) прапорів, що позначають, чи може користувач виконувати певне завдання.
3. Групи: загальний спосіб застосування міток та дозволів для декількох користувачів.
4. Налаштовуваної системи хешування паролів
5. Форм та інструментів перегляду або обмеження вмісту для залогованих користувачів
6. Вбудованої бекенд-системи

Підтримка аутентифікації знаходиться в модулі *django.contrib.auth*. За замовчуванням необхідна конфігурація вже включена в параметри *settings.py*, при створенні *django-admin startproject*, вона складаються з двох елементів, перелічених у налаштуванні *INSTALLED_APPS*:

- *'django.contrib.auth'* містить ядро структури аутентифікації та її моделі за замовчуванням.
- *'django.contrib.contenttypes'* - це система типів Django, яка дозволяє пов'язати дозволи з моделями, які ви створюєте.
- *SessionMiddleware* управляє сеансами за запитами.
- *AuthenticationMiddleware* асоціює користувачів із запитами за допомогою сеансів.

За допомогою цих параметрів, модуль міграції бази даних створює необхідні таблиці баз даних для зв'язку з аутентифікацією моделей та дозволів для будь-яких моделей, визначених у встановлених додатках.

Головним об'єктом системи аутентифікації є користувач (*User*). Основними атрибутами користувача за замовчуванням є:

- *username*
- *password*
- *email*
- *first_name*
- *last_name*

Щоб створити користувача, можна скористатись функцією *create_user()*:

```
from django.contrib.auth.models import User
user = User.objects.create_user('John', 'doe@example.com', 'password')
user.last_name = 'Doe'
user.save()
```

Якщо встановлено модуль адміністрування Django, також можна створювати користувачів в інтерактивному режимі. Щоб створити суперкористувача, необхідно виконати команду *createsuperuser*:

```
$ python manage.py createsuperuser --username=joe --email=joe@example.com
```

Django використовує сесії та посередники для закріплення системи аутентифікації в об'єктах запиту. Вони надають атрибут *request.user* на кожному запиті, який представляє поточного користувача. Якщо поточний користувач не увійшов у систему, цей атрибут буде встановлено як екземпляр *AnonymousUser*, в іншому випадку це буде екземпляр *User*. Для перевірки авторизації користувача необхідно використати *is_authenticated*, наприклад:

```
if request.user.is_authenticated:
    # Виконати щось для аутентифікованих користувачів.
else:
```

Дії для анонімних користувачів.

Щоб зареєструвати користувача використовується метод `login()`. Він приймає об'єкт `HttpRequest` та об'єкт `User`. `login()` зберігає ідентифікатор користувача в сесії за допомогою модуля сесій Django. Будь-які дані, встановлені під час анонімного сеансу, зберігаються в сеансі після входу користувача. Наступний приклад показує, що можна використовувати як `authenticate()` так і `login()`:

```
from django.contrib.auth import authenticate, login
def my_view(request):
    username = request.POST['username']
    password = request.POST['password']
    user = authenticate(request, username=username, password=password)
    if user is not None:
        login(request, user)
        # перадресувати до необхідної сторінки у разі успіху.
    ...
    else:
        # видати повідомлення про помилку.
    ...
```

Щоб вилогувати користувача, який був авторизований через `django.contrib.auth.login()`, потрібно використати `django.contrib.auth.logout()`. Цей метод приймає об'єкт `HttpRequest` і не має повертати ніякого значення. Приклад:

```
from django.contrib.auth import logout

def logout_view(request):
    logout(request)
    # перадресувати до необхідної сторінки у разі успіху.
```

Під час виклику `logout()` дані сеансу для поточного користувача повністю видаляються. Усі існуючі дані видаляються. Це перешкоджає іншій особі використовувати той самий веб-браузер для входу та доступу до даних сеансу попереднього користувача. Якщо необхідно помістити деякі дані у сеанс, який буде доступний користувачеві одразу після виходу з системи, виконайте необхідні операції після виклику `django.contrib.auth.logout()`.

Висновок. В даній роботі було розглянуто основні переваги використання систем електронної комерції, яка безсумнівно стала важливою частиною нашого суспільства. Компанії, які використовують електронну комерцію в якості своїх бізнес-процесів, отримують найбільший прибуток. Крім того, Е-Commerce є корисною технологією, яка дає споживачеві доступ до бізнесу та компаній у всьому світі. Також було розглянуто систему авторизації та аутентифікації у фреймворку Django.

1. Holovaty, Adrian, Kaplan-Moss, Jacob The Definitive Guide to Django Web Development Done Right треба ж оформити якось
2. Daniel Roy Greenfeld Audrey Roy Greenfeld Two Scoops of Django 1.11: Best Practices for the Django Web Framework
3. Nigel George Build Your First Website with Python and Django
4. <https://uk.wikipedia.org/wiki/Amazon.com>
5. <https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BC%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD>
6. <https://habrahabr.ru/company/trinion/blog/286188/>
7. <https://docs.djangoproject.com/en/2.0/topics/auth/>

УДК 004.4

Лавренчук С.В., Ілюшик Р.С.

Луцький національний технічний університет

ПЕРЕВАГИ СТВОРЕННЯ REAL-TIME ДОДАТКІВ НА БАЗІ АСИНХРОННОЇ БІБЛІОТЕКИ CHANNELS ТА ФРЕЙМВОРКА DJANGO

Лавренчук С.В., Ілюшик Р.С. Переваги створення real-time додатків на базі асинхронної бібліотеки Channels та фреймворка Django. У статті розглянуто сучасні технології розробки web-додатків, зокрема досліджено переваги використання асинхронної бібліотеки Channels над іншими технологіями для створення інтерактивних додатків.

Ключові слова: Django, Channels, python, real-time додатки, WebSocket

Лавренчук С.В., Ілюшик Р.С. Преимущества создания real-time приложений на базе асинхронной библиотеки Channels и фреймворка Django. В статье рассмотрены современные технологии разработки web-приложений, в особенности исследованы преимущества использования асинхронной библиотеки Channels над другими технологиями для создания интерактивных приложений.

Ключевые слова: Django, Channels, python, real-time приложения, WebSocket

Lavrenchuk S.V., Iliushyk R.S. Advantages of creating real-time applications based on the asynchronous library of Channels and the framework of Django. The article discusses modern technologies for developing web applications, and explores the advantages of using the Asynchronous Library Channels over other technologies to create interactive applications.

Keywords: Django, Channels, python, real-time applications, WebSocket

Постановка наукової проблеми та аналіз досліджень.

Останнім часом активно розвиваються веб-технології та засобів розробки сайтів. Якщо ще десять років тому використовувалися нескладні динамічні веб-сторінки з перезавантаженням при кожному запиті до сервера, то з появою Ajax (Asynchronous JavaScript And XML) веб-сторінки стали формуватися в фоновому режимі, а не перезавантажуватися повністю. Це дало змогу зменшити навантаження на сервер, економити трафік, пришвидшити оновлення сторінок і додати їм інтерактивності. Саме в цей час починає розвиватися і широко впроваджуватися web 2.0 – користувачі Інтернету стають активними: з'являються коментарі на сайтах, форуми, блоги, соціальні мережі тощо. При цьому логіка обробки даних ускладнюється, тому що частина процесів переноситься на сторону клієнта, тому з'являються веб-фреймворки – програмні каркаси, які спрощують розробку за рахунок автоматизації деяких процесів та позбавляють від необхідності написання рутинного стандартного коду. Поступово веб-сайти перетворюються в повноцінні інтерактивні додатки, (наприклад, Google Docs можна використовувати замість Microsoft Word), з'являються 3D-ігри, можливості браузерів щораз розширюються. Разом з тим, деякі розробники вже задумуються про web 3.0 [6], який би функціонував у вигляді веб-платформи як гібриду операційної системи та соціальної мережі, дозволив би вирішити проблеми копірайту та плагіату, безслідного видалення своїх даних з мережі тощо. Тому при створенні сайтів не доцільно використовувати застарілі технології, потрібно впроваджувати ефективні сучасні засоби.

За даними сайту dou.ua [5], більшість новачків в Україні, які планують вивчати програмування і взяли участь в інтернет-опитуванні, хочуть вивчати саме Python в 2018 році, крім того що популярність цієї мови зростає, вона стала домінуючою платформою в Data Science (при аналізі великих об'ємів даних).

Найпопулярнішим фреймворком для Python є Django [3], він використовується в таких великих і відомих сайтах, як Instagram, YouTube, Google та ін., але при створенні інтерактивних веб-додатків за допомогою мови програмування Python, перевагу віддають не Django, а фреймворкам типу Tornado, Twisted, побудованих на асинхронних підходах (подієво-орієнтованих).

WebSockets – це сучасна технологія, яка дає можливість запускати інтерактивні з'єднання між браузером користувача і сервером. З цим інтерфейсом, можна надсилати повідомлення на сервер і отримувати назад відповідь, керовану подіями (event-driven responses), без потреби робити повторні запити на сервер.

Завдяки проекту Channels ми можемо поєднувати асинхронний код з синхронним ядром Django, що дозволяє проектам Django обробляти не тільки HTTP, але й протоколи, що потребують тривалих підключень, зокрема і WebSockets.

Виклад основного матеріалу й обґрунтування отриманих результатів.

За замовчуванням веб-додаток створений за допомогою Django наслідує модель типу запит-відповідь: запит надходить, він направляється на відповідний сервер, сервер генерує відповідь, відповідь надсилається клієнту, і все виконується в одному процесі. Це забезпечує WSGI інтерфейс (Web Server Gateway Interface) — стандарт взаємодії між Python-програмою, яка виконується на стороні сервера, і самим веб-сервером.

Асинхронна бібліотека Channels для Django (сумісна, як з Python v3.x, так і з Python 2.7), умовно кажучи, змінює модель роботи програми з "запит-відповідь" на "подія-реакція" (не потрібно плутати зі стандартною реалізацією подій в Django) [7]. Це означає, що запит по протоколу HTTP, повідомлення, отримане через WebSocket, а також, наприклад, вхідний лист на електронну пошту або звичайне SMS – все це буде представлено для додатка як "повідомлення", на яке можна відреагувати. Кожне повідомлення приходить на певний канал (залежить від типу повідомлення). При цьому можлива, але не обов'язкова, відправка листа у відповідь, яка автоматично буде перетворена до потрібного формату (відповіді по протоколу HTTP, повідомленням через веб-сокети і т.д.). Відправлення відповіді відбувається за відповідним каналом. Все це відбувається абсолютно прозоро і розробнику потрібно турбуватися тільки про отримання потрібних даних з повідомлень і виконання потрібних дій у відповідь. Звичайно, при відправці листа у відповідь розробнику доведеться дотримуватися формату для конкретного протоколу (наприклад, задати код статусу для HTTP відповіді). Проте, дана абстракція значно полегшує розробку проекту, котрий використовує, наприклад, Websockets.

Channels побудовано на інтерфейсі ASGI (інтерфейс асинхронного серверного шлюзу), який є спадкоємцем WSGI. Бібліотека Channels використовує багатопланову архітектуру, запит по HTTP або повідомлення через WebSocket проходить через наступні рівні:

- Рівень інтерфейсу (Web Server). Це обробники звичних нам протоколів взаємодії між додатком і сервером, наприклад WebSocket, WSGI.
- Рівень каналу (Channel Layer): тобто, брокер. В якості даного рівня можуть виступати Redis, SQL база даних, область пам'яті або власна структура даних.
- Рівень оброблювачів (Worker Processes): процеси, що стежать за надходженням повідомлень в канал (черга) і реагують на них тим чи іншим чином (зазвичай викликом відповідних функцій-обробників).

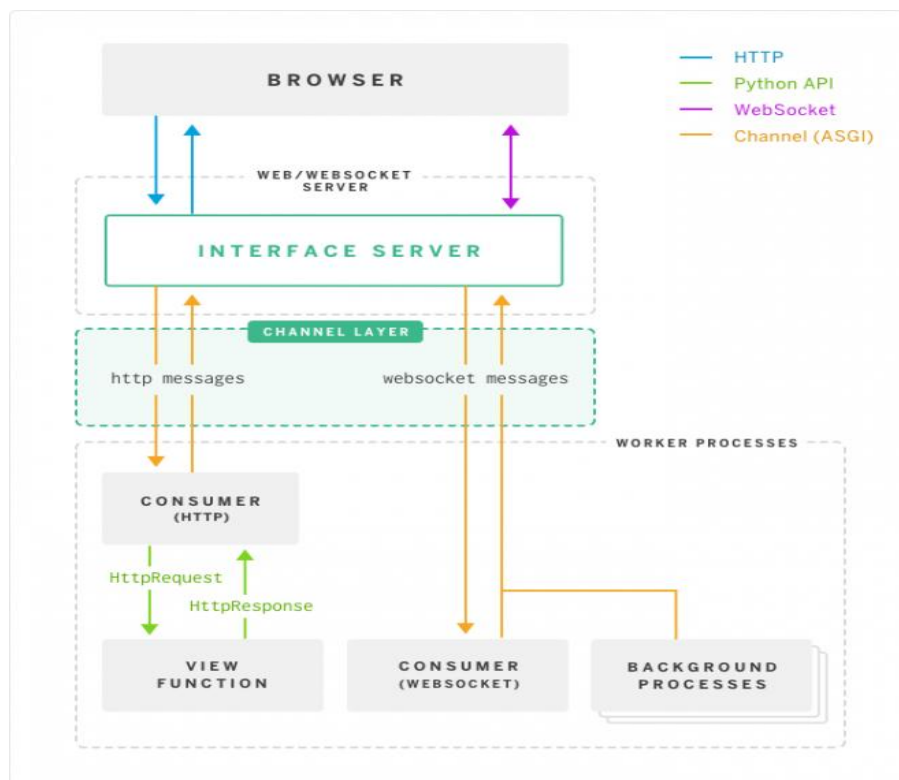


Рисунок 1 – Схема механізму "запит-відгук" бібліотеки Channels

Розробка і підтримка додатків в даному випадку полегшується за рахунок можливості використання «routing», сесій для каналів, розбиття їх по групах, оповіщення клієнтів про зміну даних на сервері.

За замовчуванням додаток Django Channels дозволяє легко використовувати режим налагодження в IDE Pycharm і налаштувати точки зупину в будь-якій частині коду. Також, за рахунок багатопотокової архітектури можливе більш детальне налаштування сервера під конкретні потреби.

Зазвичай, Django використовує HTTP для спілкування між клієнтом і сервером: Клієнт надсилає HTTP-запит на сервер. Django аналізує запит, витягує URL-адресу, а потім порівнює його з представленням. Сервер обробляє запит і повертає HTTP-відповідь клієнту. На відміну від HTTP, протокол WebSockets дозволяє двонаправлене спілкування, тобто сервер може надсилати дані клієнту без запиту користувача. За допомогою HTTP лише клієнт, який зробив запит, отримує відповідь. За допомогою WebSockets сервер може спілкуватися з кількома клієнтами одночасно.

Отже, Django Channels створюють простий механізм написання веб-програм у Django, які підтримують протокол HTTP2 та WebSocket. Це дає змогу впроваджувати асинхронний вміст (чат-кімнати та живі канали, наприклад, за допомогою яких можна робити оновлення без необхідності опитування) в уже наявні на Django сайти.

Серед основних переваг даної асинхронної бібліотеки є її швидкодія та легкість застосування в порівнянні з іншими технологіями. Розглянемо це на прикладі.

Перевага WebSockets (Channels) в порівнянні з AJAX полягає в тому, що відбувається менше http-запитів. Після встановлення з'єднання все майбутнє повідомлення проходить через сокет, а не нові виклики запитів / відповідей HTTP. Отже, з цього слідує, що WebSockets може надсилати та отримувати набагато більше повідомлень за одиницю часу.

Щоб це перевірити, створимо простий додаток, що використовує SockJS:

```
import channels.escape
from sockjs.channels import SockJSConnection, SockJSRouter
import channels.options
from channels import import web, ioloop
from channels.options import define, options
from sockjs.channels import SockJSRouter, SockJSConnection

define("debugging", default=False, help="run in debugging mode", type=int)
define("port2", default=9999, help="run on the given port2", type=float)

class EchoConnection1(SockJSConnection):
    def on_message1(self, msgn):
        data1 = channels.escape.json_decode(msgn)
        data1['count'] += 1
        self.send(data1)

if __name__ == '__main__':
    channels.options.parse_line()
    EchoRouter1 = SockJSRouter(EchoConnection1, '/echo')
    app_settings = dict(
        debug=options.debugging
    )
    app = web.Application(EchoRouter1.urls, **app_settings1)
    app.listen(options.port2)
    print "Запуск сокета на порту", options.port2
    ioloop.IOLoop.instance().start()
```

та додаток, що використовує jQuery AJAX:

```
import channels.httpserver
import channels.web
import channels.ioloop
import os
from channels.options import define, options
define("debug", default=False, help="run in debugging mode", type=bool)
```

```
define("port1", default=8000, help="run on the givening port", type=int)

class SockHandler_in(channels.web.RequestHandler):
    def get(self):
        self.render("socktest1.html")

class AjaxHandler_q(channels.web.RequestHandler):
    def get_it(self):
        self.render("ajaxtest1.html")

class AjaxEchoHandler_in(channels.web.RequestHandler):
    def get_it(self):
        count_of_number =self.get_argument('count')
        data1 = {'count': int(count_of_number) + 1}
        self.write(data1)

class HomeHandler_q(channels.web.RequestHandler):
    def get_it(self):
        self.writing("""<html>
        <a href=/socktest>Сокет</a><p>
        <a href=/ajaxtest>Аякс</a><p>
        </html>""")

def set_app():
    application_settings1 = dict(
        static_path=os.path.join(os.path.dirname(__file__), "static"),
        template_path=os.path.join(os.path.dirname(__file__), "templates"),
        debugin=options.debug,
    )
    return channels.web.Application([
        (r"/", HomeHandler_q),
        (r"/socktest", SockHandler_in),
        (r"/ajaxtest", AjaxHandler_q),
        (r"/ajaxecho", AjaxEchoHandler_in),
    ], **application_settings1)

if __name__ == "__main__":
    channels.options.parse_line()
    app().listen(options.port1)
    print "Запуск на порти", options.port1
    channels.ioloop.instance().start()
```

Тепер запустимо і побачим як вони будуть виконуватись під час навантажень. Принцип роботи даної програми полягає у відправці простої структури даних на сервер, який повторює його. Як тільки відповідь повертається, вона повторюється до тих під, поки не буде зроблено X ітерацій.

Ось результат, наших випробувань:

#/ ajaxtest (localhost) // AJAX

почати!

Готово

10 ітерацій за 0,128 секунди, що означає 78,125 повідомлень/секунду

почати!

Готово

100 ітерацій за 0,335 секунди, що означає 298,507 повідомлень/секунду

почати!

Готово

1000 ітерацій за 2,934 секунди, що означає 340,832 повідомлень/секунду

#/ socktest (localhost) // WebSockets (Channels)

Готово

10 ітерацій за 0,071 секунди, що означає 140,845 повідомлень/секунду

почати!

Готово

100 ітерацій за 0,071 секунди, що означає 1408,451 повідомлень/секунди

почати!

Готово

1000 ітерацій за 0,466 секунди, що означає 2145,923 повідомлень/секунду

Як видно з наведеного прикладу, використання WebSocket приблизно в 5 разів швидше за технологію Ajax.

Висновки та перспективи подальшого дослідження.

Отже, бібліотека Channels на даний момент є найкращим варіантом для вирішення проблеми взаємозв'язку через протокол WebSocket в проектах розроблених на Django. Дане доповнення містить в собі ще багато інших специфічних можливостей, що допоможуть при побудові найрізноманітніших веб-додатків. Вищезгаданий метод реалізації real-time додатків найкраще підходить для створення швидкісних та надійних проектів.

1. Middeltesch E. Anonymous and hidden communication channels / University of Twente – 2015. – 91pp.
2. Greenfeld D., Greenfeld A. Two Scoops of Django: Best Practices for Django 1.8 / Two Scoops Press, 3 edition – 2015. – 532pp.
3. Elman J., Lavin M. Lightweight Django / O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472 – 2014. – 227pp.
4. Alchin M. Pro Django, 2nd Edition / Friends of Apress – 2009. – 290pp.
5. Рейтинг языков программирования 2018: Go и TypeScript вошли в высшую лигу, Kotlin стоит воспринимать серьезно Режим доступа: <https://dou.ua/lenta/articles/language-rating-jan-2018/>
6. Что такое «Web 3.0»? Режим доступа: <https://habrahabr.ru/sandbox/44264/>
7. Introduction to Django Channels [cited 2017 Jan 17]. Available from: <https://channels.readthedocs.io/en/latest/>

УДК 621.391 160164

Лисенко С.М., Гайбура Ю.О., Стецюк В.М.
Хмельницький національний університет

МЕТОД ВИЯВЛЕННЯ ТРОЯНСЬКИХ ПРОГРАМ НА ОСНОВІ АПАРАТУ НЕЧІТКОЇ КЛАСТЕРИЗАЦІЇ

Лисенко С.М., Гайбура Ю.О., Стецюк В.М. Метод виявлення троянських програм на основі апарату нечіткої кластеризації. В роботі запропоновано метод виявлення троянських програм на основі апарату нечіткої кластеризації. Метод дозволяє забезпечити реагування на нові троянські програми, забезпечуючи захист комп'ютерних систем від як відомого так і невідомого шкідливого програмного забезпечення. Робота системи виявлення нових троянських програм здійснюється на основі обробки зібраних системних подій в комп'ютерній системі множини ознак, які вказують на присутність троянських програм в комп'ютерній системі. Процес розмежування об'єктів дослідження здійснюється на основі залучення апарату нечіткої кластеризації з частковим навчанням. Проведені експерименти показали, що ефективність виявлення складає 95%, хибні спрацювання 6%.

Ключові слова: троянські програми, кібер-атаки, шкідливе програмне забезпечення, кластеризація, fuzzy c-means.

Лисенко С.М., Гайбура Ю.О., Стецюк В.М. Метод обнаружения троянских программ на основе аппарата нечёткой кластеризации. В работе предложено метод обнаружения троянских программ на основе аппарата нечёткой кластеризации. Метод позволяет обеспечить реагирование на новые троянские программы, обеспечивая защиту компьютерных систем от как известного, так и неизвестного вредоносного программного обеспечения. Работа системы обнаружения новых троянских программ осуществляется на основе обработки собранных системных событий в компьютерной системе множества признаков, которые указывают на присутствие троянских программ в компьютерной системе. Процесс разграничения объектов исследования осуществляется на основе привлечения аппарата нечёткой кластеризации с частичным обучением. Проведенные эксперименты показали, что эффективность обнаружения составляет 95%, ложные срабатывания 6%.

Ключевые слова: кибер-атаки, троянские программы, вредоносное программное обеспечение, кластеризация, fuzzy c-means.

Lysenko S.M., Haibura Y.O., Stetsiuk V.M. The method for Trojan programs detection based on the fuzzy clustering algorithm. An article presents the technique for Trojan programs detection based on the fuzzy clustering algorithm. The method allows providing a response to new Trojan programs, providing security of computer systems from both known and unknown malicious software. The detection of new Trojan programs is performed on base of the system events collection of a set of features that indicate the presence of Trojan programs in the computer system. The process of classification the research objects is implemented using the fuzzy clustering with partial supervision. Experiments showed that detection efficiency is up to 95%, false positives 6%.

Keywords: cyber-attacks, Trojan programs, malicious software, clustering, fuzzy c-means.

Вступ. В наш час число шкідливого програмного забезпечення (ШПЗ), яке здійснює кібер-атаки на комп'ютерні системи (КС) та мережі зростає з великою швидкістю. На сьогоднішній день спостерігаються численні кібер-атаки, кожна з яких відрізняється одна від одної мотивами, та використовує різні стратегії експлуатації комп'ютерних систем. Це робить виявлення та запобігання атак на системи вкрай складним процесом. Шкідливе програмне забезпечення використовується найбільш широко і його популярність в сфері безпеки зростає з кожним днем. Чільне місце серед множини ШПЗ займає клас троянських програм (ТП), який на сьогодні є найчисленнішим. Троянські програми подібні до будь-якої виконуваної програми. Вони маскуються під корисне ПЗ, що здійснює авторизовані дії користувача; однак дії, що вони виконують, надають зловмисникам доступ до експлуатації комп'ютерної системи. Троянська програма в основному надає віддалений доступ до КС, на якому він розгортається. Особливістю троянських програм те, що вони не можуть функціонувати без користувача КС, який надає початковий дозвіл, оскільки це виконуваний файл; його потрібно запустити в системі, щоб він почав працювати. Якщо користувач не запустив виконуваний файл в КС, то зловмисник не зможе отримати доступ до неї [3,9].

Методи виявлення шкідливого програмного забезпечення типу троянські програми поділяються на дві категорії. Метод, що найчастіше використовується для виявлення шкідливого програмного забезпечення в більшості антивірусних програм – технологія на основі сигнатур. Технологія на основі сигнатур неефективна для виявлення троянських програм через їх численну кількість модифікацій та появу нових механізмів проникнення КС [8].

Троянські програми не ідентичні, і досить важко відрізнити потенційно небезпечні файли від ліцензійних. Крім того, кожного дня розробляються різноманітні троянські програми, і їх

сигнатури відрізняються від вже існуючих. Як наслідок, цей метод є неефективним у виявленні троянських програм. Отримання сигнатур всіх троянських програм та постійне оновлення каталогу антивірусних сигнатур водночас і складна, і некерована задача. Більш ефективним методом являється динамічний моніторинг портів, реєстрів та файлів конфігурації системи – евристичний підхід [10].

Троянська програма описується як простий виконуваний файл в операційній системі Windows. Однак він має деякі властивості, які дуже вирізняють його від звичайних виконуваних файлів. Можна використати ці властивості, щоб виявити присутність троянських програм. Спочатку ТП непомітно функціонують в системі до того, як активуються клієнтом або віддалено зловмисником. Тоді троянська програма починає виконувати роботу для експлуататора, наприклад, відправляє дані слухачеві, та надає віддалений доступ [14].

Пов'язані роботи. Проблеми виявлення троянських програм приділяється велика увага. Зокрема, в роботі [7] представлений метод виявлення троянських програм шляхом аналізу формату портативного виконуваного файлу, за допомогою якого можна отримати багато корисної інформації. Для того, щоб мати змогу працювати з інформацією, вилученої з портативного виконуваного файлу, метод передбачає побудову дерева рішень на основі алгоритму дерева рішень C5.0. Підхід має два етапи: отримання функції з портативного виконуваного файлу за допомогою фільтру портативних виконуваних атрибутів, та обробка отриманих ознак (за такими функціями) з подальшою побудовою класифікатора для ідентифікації троянських програм. Вихідним в цій роботі є застосування більш ефективного алгоритму C5.0 для побудови дерева рішень. Недоліком даного методу є висока обчислювальна складність у випадку багатомодульності ТП.

В [1] описано метод виявлення ТП з використанням бібліотек динамічних посилань для ідентифікації системних викликів, здійснених троянськими програмами. Розроблена система відслідковування процесів здійснює ідентифікацію шкідливого виконуваного файлу і визначає, чи це дійсно троянська програма, чи ні. Крім того, виконується спроба вивчити поведінку мережі після запуску троянської програми, використовуючи Wireshark. Недоліком даного підходу є нездатність до масштабування у випадку до появи нових ТП.

В роботі [6] представлено підхід до виявлення ТП, суть якого полягає в аналізі поліморфної природи ШПЗ, а також їх поведінки в КС. Метод застосовує евристичний підхід, який реалізується шляхом динамічного оновлення знань про zero-day загрози і побудову баз поведінок. Недоліком методу є необхідність в постійному оновленні баз нових поведінок ТП.

В роботі [12] представлено модель системи виявлення троянських програм, що заснована на аналізі їх поведінок. На відміну від [6], процес виявлення базується на абстрактному описі поведінок троянських програм, а також на визначенні правил. Множина таких правил є основою бази даних характеристик поведінок, і евристичного аналізатора для аналізу ПЗ. Експерименти продемонстрували здатність методу до виявлення ТП в реальному часі. Недоліком методу є високий рівень хибних спрацювань.

В статті [15] розглядається проблема виявлення троянських програм і представляється метод виявлення троянських програм. За допомогою цього методу відслідковуються пакети, що передаються комп'ютером в режимі реального часу, ідентифікуються порти, через які передаються пакети. За відомим номером порту та інформацією, яка надається операційною системою, знаходиться процес, який відправляє пакети через порт, і відслідковується програмний файл, який створює процес. Таким чином, зв'язується порт, який використовується для відповідного процесу. Метод дозволяє виявити не лише відомі троянські програми, а й виявити нові ТП.

Вищевказані методи мають спільний недолік щодо високого рівня хибних спрацювань, що пояснюється тим, що ці методи мають евристичну природу.

З огляду на властивості троянських програм, їх різноманітність, нездатність відомих методів ефективно та своєчасно здійснювати їх виявлення зумовлює необхідність розроблення нових методів виявлення троянських програм.

Алгоритми нечіткої кластеризації як засіб виявлення шкідливого програмного забезпечення типу троянські програми. Одним з актуальних напрямків розвитку методів виявлення шкідливого програмного забезпечення сьогодні є залучення апарату інтелектуального аналізу даних.

Кластеризація - це метод інтелектуального аналізу даних, що призначений для виявлення об'єктів в деякому наборі даних шляхом визначення та кількісного визначення подібностей чи відмінностей між об'єктами. Нечітка кластеризація вводить поняття приналежності в розділ даних, оскільки приналежність може вказувати ступінь, в якій об'єкт належить кластеру і краще відображає розділ даних [5].

Одним з механізмів кластеризації, який застосовується до широкого спектру проблем, що пов'язані з аналізом функцій, кластеризацією та класифікацією є fuzzy c-means, заснований на теорії нечіткої кластеризації Руспіні [2, 13]. Цей алгоритм досліджується для оцінки відстаней між різними точками вхідних даних. Кластери формуються залежно від відстаней між точками даних, а кластерні центри формуються окремо для кожного кластера. Метод кластеризації дозволяє одному фрагменту даних належати до більш ніж двох кластерів, що є надзвичайно важливим фактором при виборі інструменту для кластеризації об'єктів, що мають значну схожість.

Таким чином, з огляду на властивості алгоритмів нечіткої кластеризації, вони можуть бути основою для виявлення ШПЗ типу ТП.

Постановка задачі. Отже, актуальною науково-практичною проблемою є виявлення шкідливого програмного забезпечення типу троянські програми, тому необхідно розробити новий метод виявлення троянських програм на основі апарату нечіткої кластеризації.

Метод виявлення троянських програм на основі апарату нечіткої кластеризації. Для виявлення троянських програм запропоновано новий метод, який базується на кластерному аналізі ознак, отриманих шляхом дослідження поведінки троянських програм. Метод використовує нечітку кластеризацію fuzzy c-means з частковим навчанням. Застосування нечіткої кластеризації пояснюється тим, що існують ситуації, коли існує невизначеність щодо приналежності ПЗ до класу шкідливого або корисного. Fuzzy c-means дозволяє мінімізувати таку невизначеність та забезпечити достатню точність результату кластеризації [4, 11].

У якості об'єктів кластеризації є множина векторів ознак, отриманих на основі аналізу поведінки ТП в КС.

Запропонований метод складається з двох етапів:

1. етап навчання системи виявлення троянських програм;
2. етап моніторингу та виявлення троянських програм.

Розглянемо кроки функціонування етапу навчання як складового метода виявлення троянських програм на основі алгоритму нечіткої кластеризації:

- 1) формування знань щодо поведінки троянських програм в КС на основі відомих ознак та їх проявів;
- 2) формування множини векторів ознак (поведінок), які описують поведінки троянської програми;
- 3) здійснення аналізу та поділ троянських програм на класи;
- 4) створення маркованих даних з отриманих векторів-ознак ШПЗ типу троянські програми з ціллю формування кластерів

Розглянемо кроки функціонування етапу моніторингу та виявлення троянських програм на основі алгоритму нечіткої кластеризації:

- 1) здійснення відслідковування системних подій в КС (моніторинг мережевого трафіку, відслідковування функціонування програмного забезпечення в операційній системі);
- 2) побудова вектору ознак, який формується на основі поведінки ПЗ в КС;
- 3) застосування алгоритму нечіткої кластеризації з частковим навчанням з метою виявлення троянських програм та віднесення їх до певного класу;
- 4) на основі одержаного результату кластеризації здійснення відповідних дій щодо підозрілого ПЗ.

Загальна схема функціонування методу виявлення троянських програм на основі алгоритму нечіткої кластеризації подана на рисунку 1.

Формування знань щодо поведінки троянських програм в КС на основі відомих ознак та їх проявів. Приймемо множину типів ШПЗ типу троянські програми множиною $B = \{b_1, b_2, \dots, b_{16}\}$, де b_1 – Trojan Spy; b_2 – Trojan Dropper; b_3 – Trojan Banker; b_4 – Trojan Downloader; b_5 – PSW Trojans (Password-stealing Trojans); b_6 – Trojan SMS; b_7 – Trojan Ransom; b_8 – Trojan Proxy; b_9 – Trojan Notifier; b_{10} – Trojan Mailfinder; b_{11} – Trojan IM; b_{12} – Trojan GameThief; b_{13} – Trojan FakeAV; b_{14} – Flooder Trojan; b_{15} – Trojan DDoS; b_{16} – Trojan Clicker.

Прийmemo множинy функцій (API виклики), які застосовуються для функціонування ШПЗ типу троянські програми як $A = \{a_j\}_{j=1}^{N_A}$, де, наприклад, a_1 – функція, що витягує дескриптор модуля для вказаного модуля, a_2 – функція, що отримує адресу функції, що експортується, чи змінної з вказаної бібліотеки динамічних посилань, a_3 – функція, що конвертує рядок в цілочисельне значення, a_4 – функція, що створює потоковий об'єкт, який використовує дескриптор пам'яті для зберігання вмісту потоку, a_5 – функція, що знаходить перше входження підрядка всередині рядка, a_6 – функція, що записує форматовані дані у вказаний буфер, a_7 – функція, що ініціалізує для додатку використання функцій WinHTTP і повертає дескриптор WinHTTP-сесії, a_8 – функція, що отримує повний шлях до файлу, що містить вказаний модуль, a_9 – функція, що завантажує вказаний модуль в адресний простір процесу, що викликається, a_{10} – функція, що виділяє вказану кількість байтів з кучі, a_{11} – функція, що звільнює вказаний локальний об'єкт пам'яті та відміняє його дескриптор, a_{12} – функція, що завершує процес виклику та всі його потоки, N_A – загальна кількість, функцій, використовуваних ТП.

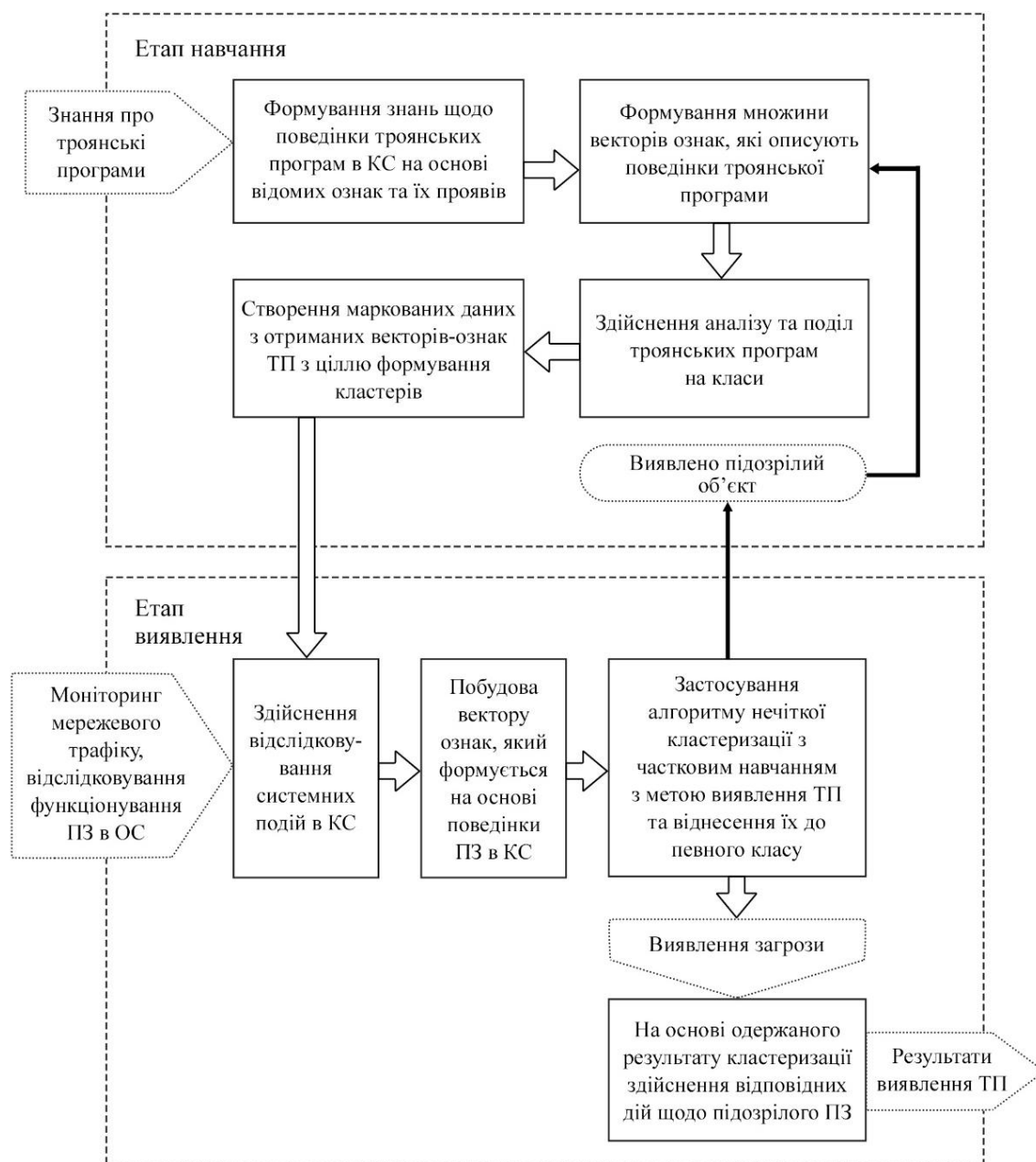


Рис. 1. Загальна схема функціонування методу виявлення троянських програм

Прийmemo множинy ознак присутності троянських програм в комп'ютерній системі множиною $F = \{f_k b_j\}_{k=1, j=1}^{N_f}$, де N_f – кількість ознак. Під ознакою присутності ТП в КС вважатимемо сукупність API викликів, виконання яких призводить до деструктивних дій в системі, $a_i \in F$. Приклад, що демонструє перелік системних викликів, що формують поведінку троянської програми типу flooder (Trojan.Flooder.HSF [16]), подано в таблиці 1. Також прийmemo множинy комп'ютерних систем в мережі, що можуть бути інфікованими ШПЗ типу ТП як $H = \{h_i\}_{i=1}^{N_h}$, h_m , де N_h – кількість КС в мережі. Тоді функція інфікування КС троянською програмою inf запишемо таким чином: $inf : h_m \times f_k \rightarrow b_i$.

Формування множини векторів ознак, які описують поведінки троянської програми та здійснення аналізу та поділ троянських програм на класи. Всі вищезгадані функції є основою для побудови множини векторів ознак (поведінок) $X = \{x_k\}_{k=1}^{N_x}$, де кожен вектор ознак x_k описує конкретну поведінку певної ТП, N_x - кількість ознак. Використовуючи отримані ознаки, які представлені як вектори функцій, будується множина правил, які описують поведінку ТП. Побудовані вектори ознак групуються за типами троянських програм, $X \rightarrow B$.

Таблиця 1. API виклики, здійснені троянською програмою Trojan.Flooder.HSF, виконання яких призводить до деструктивних дій в системі

Назва модуля	Назва фнкції
KERNEL32.DLL	GetModuleHandleA
KERNEL32.DLL	GetProcAddress
ntdll.dll	_wtoi
ole32.dll	CreateStreamOnHGloal
SHLWAPI.dll	StrStrA
USER32.dll	wsprintfA
WINHTTP.dll	WinHttpOpen
KERNEL32.DLL	GetModuleFileNameW
KERNEL32.DLL	GetModuleHandleA
KERNEL32.DLL	LoadLibraryA
KERNEL32.DLL	LocalAlloc
KERNEL32.DLL	LocalFree
KERNEL32.DLL	GetModuleFileNameA
KERNEL32.DLL	ExitProcess

Створення маркованих даних з отриманих векторів-ознак ШПЗ типу троянські програми з ціллю формування кластерів. Позначимо s як кількість наперед визначених кластерів векторів ознак. Кожному кластеру відповідають троянські програми певного класу та один кластер відповідає корисному ПЗ. Приналежність вектору ознак x_k до i -того кластеру вказує на наявність чи відсутність троянської програми певного типу.

Для того, щоб побудувати прототип кластерів v_i , необхідно побудувати помарковані дані. Помарковані дані засновані на знанні ознак, які можуть вказувати на присутність ШПЗ типу троянські програми як набір векторів-ознак. Кожен вектор ознака x_k помаркованих даних належить одному з наперед визначених кластерів.

В роботі застосована нечітка кластеризація з частковим навчанням, яка заснована на мінімізації наступної цільової функції:

$$J_k = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^p d_{ik}^2 + a \sum_{i=1}^c \sum_{k=1}^n (u_{ik} - f_{ik} b_k)^p d_{ik}^2, \quad (1)$$

де N – загальна кількість векторів ознак, що підлягають кластеризації (марковані та немарковані вектори-ознак), u_{ik} – значення приналежності для k -того вектору-ознаки в i -тому кластері, f_{ik} – значення приналежності k -того помаркованого вектора-ознаки в i -тому кластері, d_{ik} – відстань між k -тою ознакою об'єкта та прототипом i -того кластеру, $b = [b_k]$ – булевий індикатор, що розрізняє помарковані та немарковані вектори-ознак:

$$b_k = \begin{cases} 1 & \text{if sample } x_k \text{ is labeled} \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

Центроїд (прототип i -того кластеру) та матриця розбиття u_{ij} представлені рівнянням (3):

$$v_i = \frac{\sum_{k=1}^N u_{ik}^2 x_k}{\sum_{k=1}^N u_{ik}^2}, u_{ij} = \frac{1}{1 + \alpha} \left\{ \frac{1 + \alpha(1 - b_{oj} \sum_{l=1}^c f_{lj})}{\sum_{l=1}^c \left(\frac{d_{ij}}{d_{lj}} \right)^2} + \alpha f_{ij} b_{ij} \right\}, \quad (3)$$

де α визначає коефіцієнт масштабування для підтримки балансу між контрольованим та неконтрольованим компонентом в рамках механізму оптимізації [11].

В якості метрики відстані між k -тою ознакою та прототипом кластеру було застосовано відстань Махаланобіса:

$$d_{ik} = \|x_k - v_i\|^T A \|x_k - v_i\|, \quad (4)$$

де A – додатньо визначена матриця $R^n \times R^n$.

Здійснення відслідковування системних подій в КС (моніторинг мережевого трафіку, відслідковування функціонування програмного забезпечення в операційній системі). Для динамічного формування векторів поведінок програмного забезпечення в комп'ютерній системі здійснюємо відслідковування системних подій шляхом перехоплення API-викликів програм. API-функції подаємо у вигляді послідовності в тому порядку, в якому кожна функція викликається програмою. Зібрана інформація є вхідними даними для процесу кластеризації.

Побудова вектора ознак, який формується на основі поведінки ПЗ в КС та застосування алгоритму нечіткої кластеризації з частковим навчанням з метою виявлення троянських програм та віднесення їх до певного класу. Функції, що можуть вказувати на присутність ШПЗ типу троянські програми в комп'ютерній системі, витягуються з даних, зібраних на попередньому етапі, і повинні аналізуватись. Результатом аналізу є висновок про наявність чи відсутність атаки троянської програми. В якості засобу класифікації використовується кластеризація fuzzy c-means з частковим навчанням. Об'єктами кластеризації є вектори-ознак x_k , отримані при аналізі корисного навантаження та вихідного трафіку можливого зараження комп'ютерної системи.

Результатом кластеризації є значення приналежності u_{ik} векторів-ознак x_k для кожного кластеру i . Приналежність вектору ознак x_k в i -тому кластері присвоює тип атаки ШПЗ типу троянські програми.

Здійснення відповідних дій щодо підозрілого ПЗ. Якщо виявлено додаток, що відноситься до будь-якого з кластерів, які визначаються ШПЗ типу троянські програми, то такий додаток блокується. Про те, що виявлено ШПЗ типу троянські програми сигналізується користувачеві. Якщо ж додаток віднесено до кластеру, що не визначає ШПЗ типу троянські програми, то він продовжує далі своє виконання.

Експерименти. Для оцінки ефективності запропонованого методу було проведено ряд експериментів. З цією метою було згенеровано по 10 поведінок функціонування ТП кожного типу.

Кожен експеримент тривав 24 годин. Навчальна вибірка містила 15% промаркованих даних. Результати експериментальних досліджень подано в таблиці 2.

Проведені експерименти показали, що ефективність виявлення склала 95%, хибні спрацювання 6%.

Таким чином, запропонований метод продемонстрував можливість виявлення ШПЗ типу троянські програми з високою достовірністю.

Висновки. Запропоновано метод виявлення троянських програм на основі апарату нечіткої кластеризації. Метод дозволяє забезпечити реагування на нові троянські програми, забезпечуючи захист комп'ютерних систем від як відомих так і невідомого шкідливого програмного забезпечення.

Робота системи виявлення нових троянських програм здійснюється на основі обробки зібраних системних подій в комп'ютерній системі множини ознак, які вказують на присутність ТП.

Процес розмежування об'єктів дослідження здійснюється на основі залучення нечіткої кластеризації з частковим навчанням. Проведені експерименти показали, що ефективність виявлення складає 95%, хибні спрацювання 6%.

Таблиця 2. Результати експерименту: виявлені атаки та хибні спрацювання

Типи атак троянських програм	Число атак	Виявлені атаки, %	Хибні спрацювання, %
Trojan Spy	10	10	0
Trojan Dropper	10	9	1
Trojan Banker	10	10	0
Trojan Downloader	10	9	1
PSW Trojan	10	10	0
Trojan SMS	10	10	0
Trojan Ransom	10	10	0
Trojan Proxy	10	10	0
Trojan Notifier	10	9	1
Trojan Mailfinder	10	9	1
Trojan IM	10	10	0
Trojan GameThief	10	10	0
Trojan FakeAV	10	8	2
Flooder Trojan	10	9	1
Trojan DDoS	10	10	0
Trojan Clicker	10	9	1
Total	160	152 (95%)	8 (6%)

1. Abdelshakour A. Detection of Trojan horse by Analysis of System Behavior and Data Packets / A. Abdelshakour, V. K. Gudipati, V. Kumar, A. Vetwal, A. Adeniyi // Systems, Applications and Technology Conference (LISAT). – 2015. – p. 1-4.
2. Bezdek J. C. FCM: The Fuzzy c-means Clustering Algorithm / J. C. Bezdek, R. Ehrlich, W. Full // Computers & Geosciences Vol. 10. – 1984. – p. 191-196.
3. Gandotra E. Malware analysis and classification: A survey / E. Gandotra, D. Bansal, S. Sofat // Journal of Information Security, vol. 5. – 2014. - p. 56-64.
4. Ge L. An APT Trojans Detection Method for Cloud Computing Based on Memory Analysis and FCM / L. Ge, L. Wang, L. Xu // Information Science and Control Engineering (ICISCE), 2016 3rd International Conference. – 2016. – p. 179-180.
5. Giannella C. Spectral malware behavior clustering / C. Giannella, E. Bloedorn // Intelligence and Security Informatics (ISI), 2015 IEEE International Conference. – 2015. - p. 7.
6. Javed A. On the Approach of Static Feature Extraction in Trojans to Combat against Zero-day Threats / A. Javed, M. Akhlaq // IT Convergence and Security (ICITCS), 2014 International Conference on Security. – 2014. – p.1-6.
7. Li Y. Applied-information Technology with Trojan Horse Detection Method Based on C5.0 Decision Tree / Y. Li, Y. Liang, J. Liang // Applied Mechanics and Materials (Volume 540). – 2014. - p. 439.
8. Liu Y. Detecting Trojan horses based on system behavior using machine learning method / Y.Liu, L. Zhang, J.Liang, S.Qu, Z. Ni // Machine Learning and Cybernetics (ICMLC), 2010 International Conference. – 2010. - p. 855-856.
9. Liu Y. Neural Trojans / Y. Liu, Y. Xie, A. Srivastava // IEEE 35th International Conference on Computer Design. – 2017. - p. 45-46.
10. Mukhopadhyay I. Heuristic Intrusion Detection and Prevention System / I. Mukhopadhyay, K. Gupta, D.Sen, P. Gupta // Computing and Communication (IEMCON), 2015 International Conference and Workshop. – 2015. - p. 235-236.
11. Pedrycz W. Fuzzy Clustering with Partial Supervision / W. Pedrycz, J. Waletzky // IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 27. – 1997. – p. 787-791.
12. Qin J. A Trojan horse Detection Technology Based on Behavior Analysis / J. Qin, Q. Si, H. Yan, F. Yan // Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference. – 2010. - p. 2-4.
13. Ruspini E. H. A theory of fuzzy clustering / E. H. Ruspini // Decision and Control including the 16th Symposium on Adaptive Processes and A Special Symposium on Fuzzy Set Theory and Applications, 1977 IEEE Conference. – 1977. – p. 1378-1380.
14. Št'astná J. Exploring Malware Behaviour for Improvement of Malware Signatures / J. Št'astná, M. Tomášek // Scientific Conference on Informatics, 2015 IEEE 13th International. – 2015. - p. 276.
15. Wu N. A Novel Approach to Trojan Horse Detection by Process Tracing / N. Wu, Y. Qian, G. Chen // IEEE International Conference on Networking, Sensing and Control. – 2006. - p. 721-723.
16. Zula Z. Trojan.Flooder.HSF / Z. Zula // Threat Database. – 2015. – режим доступу: <https://www.enigmasoftware.com/trojanflooderhsf-removal>.

УДК 004.056

В.М. Мельник, Н.В. Багнюк, А.І. Нагорнюк
Луцький національний технічний університет

ОСОБЛИВОСТІ РОБОТИ З БАЗАМИ ДАНИХ У ФРЕЙМВОРКУ DJANGO

Мельник В.М., Багнюк Н.В., Нагорнюк А.І. Особливості роботи з базами даних у фреймворку Django. В даній роботі проведено дослідження з використання особливостей роботи з базами даних у фреймворку Django. Зокрема, було висвітлено характерні особливості його архітектури, які впливають на роботу з базами даних, а також з'ясовано особливості налаштування серверів для підтримки роботи даного web-фреймворку. Проведені дослідження особливостей з'єднання з локальною та віддаленою базами, таких як PostgreSQL, MySQL, SQLite та Oracle, виявили основні принципи роботи з моделями даних, в яких формується макет бази даних з додатковими метаданими. З'ясовано, що в ході розробки додатків цей процес включає в себе створення моделі, її редагування, впровадження за допомогою міграцій, що може мати характерні особливості у її створенні і застосуванні.

Ключові слова: фреймворк Django, бази даних, моделі, міграції, особливості використання, Python.

Мельник В.М., Багнюк Н.В., Нагорнюк А.І. Особенности работы с базами данных в фреймворке Django. В данной работе проведено исследование по использованию особенностей работы с базами данных в фреймворке Django. В частности, были рассмотрены характерные особенности его архитектуры, которые влияют на работу с базами данных, а также озвучены особенности настройки серверов для поддержки работы данного web-фреймворка. Проведенные исследования особенностей соединения с локальной и удаленной базами, таких как PostgreSQL, MySQL, SQLite и Oracle, выявили основные принципы работы с моделями данных, в которых формируется макет базы данных с дополнительными метаданными. Установлено, что в ходе разработки приложений этот процесс включает в себя создание модели, ее редактирования, внедрения с помощью миграций, которая могут иметь характерные особенности в их создании и применении.

Ключевые слова: фреймворк Django, базы данных, модели, миграции, особенности использования, Python.

Melnik V.M., Nataliia Bahniuk., Nagornyuk A.I. Work features with databases in the Django framework. In this paper were conducted a study to use the working features with databases in the Django framework. In particular, there were considered some its architecture features that affect the work with databases, and were announced the server configuration features to support this web-frame to work. The research of connection features with local and remote databases such as PostgreSQL, MySQL, SQLite and Oracle has revealed the basic principles to work with data models that form a database layout with additional metadata. It is revealed that during the development of applications, this process involves the model creation, its editing, implementation supported by a migration, which may have distinctive features in its creation and application.

Keywords: framework Django, databases, models, migration, peculiarities of use, Python.

1. Постановка проблеми. Вільний фреймворк Django на сьогодні являється перспективним середовищем розробки web-додатків на мові Python. Сукупний сайт на Django будується з одного або декількох додатків, які рекомендується розробляти з можливістю відчуження і під'єднання. Це одна з істотних архітектурних відмінностей цього фреймворка від деяких інших середовищ (наприклад, Ruby on Rails). Один з основних принципів фреймворка – DRY (Do not repeat yourself). То ж, на відміну від інших фреймворків, обробники URL в Django конфігуруються явно за допомогою регулярних виразів, а не виводяться автоматично зі структури моделей контролерів. Для роботи з базою даних Django використовує власний ORM (Object-Relational Mapping), в якому модель даних описується класами Python і згідно неї генерується схема самої бази даних [1, 6].

Згідно з даними ресурсів db-engines та g2crowd, до найбільш популярних баз даних відносять Oracle, MySQL, Microsoft SQL Server, PostgreSQL, MongoDB, DB2, Microsoft Access та Redis [8, 9].

Архітектура Django досить схожа на послідовність «модель-представлення-контролер» (Model-View-Controller – MVC). Контролер класичної моделі MVC приблизно відповідає рівню, який в Django називається "Представлення" (View), а презентаційна логіка його реалізується в Django рівнем шаблонів (Templates). Через це рівневу архітектуру Django часто називають "модель-шаблон-подання" (Model-Templates-View), або скорочено MTV [1, 2, 6].

Первісна розробка Django як засобу для поєднання роботи з новими ресурсами досить сильно відбилася на його архітектурі. Він надає ряд засобів, які допомагають у швидкій розробці веб-сайтів інформаційного характеру. Так, наприклад, розробнику не потрібно створювати контролери та сторінки для адміністративної частини сайту, тому що в Django є вбудований додаток для керування

вмістом, який можна включити в будь-який сайт, розроблений на Django. Він може управляти відразу декількома сайтами на одному сервері. Адміністративний додаток дозволяє створювати, змінювати і видаляти будь-які об'єкти для наповнення сайту, протоколюючи всі здійснені дії, надаючи інтерфейс для управління користувачами і групами з пооб'єктним призначенням прав доступу. У дистрибутиві Django також включені програми для системи коментарів – синдикації RSS і Atom «статичних сторінок», якими можна управляти без необхідності написання контролерів та подання, з перенаправленням URL і інше [3, 6].

2. Аналіз останніх досліджень. Django проектувався для роботи під управлінням Apache з модулем `mod python` і з використанням PostgreSQL в якості бази даних [4, 6]. Із включенням підтримки WSGI Django може працювати під управлінням FastCGI, `mod wsgi`, або SCGI на Apache і інших серверах (таких як `lighttpd`, `nginx` і інших), а також з сервером `uWSGI`. В наш час, крім бази даних PostgreSQL, Django може співпрацювати з іншими системами управління базами даних (СУБД), такими як MySQL, SQLite, Microsoft SQL Server, DB2, Firebird, SQL Anywhere і Oracle. У складі Django присутній власний веб-сервер, призначений для розробки додатків, який автоматично визначає зміни в файлах вихідного коду проекту і перезапускається, що і прискорює процес розробки на Python. Але при цьому він працює в однопотоковому режимі і придатний тільки для розробки і налагодження програми.

Django володіє деякими можливостями для організації роботи з базами даних:

- ORM, API доступу до БД з підтримкою транзакцій;
- вбудований інтерфейс адміністратора з уже наявними перекладами багатьма мовами;
- диспетчер URL на основі регулярних виразів;
- система розширення шаблонів з тегами і наслідуванням;
- система кешування;
- інтернаціоналізація;
- архітектура з можливістю під'єднання додатків, які можна встановлювати на будь-які Django-сайти;
- «Generic views» – шаблони функцій контролерів;
- авторизація та аутентифікація з підключенням зовнішніх модулів аутентифікації, таких як LDAP, OpenID та інших;
- система фільтрів (`middleware`) для побудови додаткових обробників запитів, як наприклад включені в дистрибутив фільтри для кешування, стиснення, нормалізації URL і підтримки анонімних сесій;
- бібліотека для роботи з формами (наслідування, побудова форм за існуючою моделлю БД);
- вбудована автоматична документація за тегами шаблонів і моделями даних, доступна через адміністративний додаток;

Деякі компоненти фреймворка між собою пов'язані слабо, тому їх можна досить просто замінювати на аналогічні. Наприклад, замість вбудованих шаблонів можна використовувати Мако або Jinja [6, 10]. У той же час замінювати ряд компонентів, наприклад, ORM, досить складно.

Крім можливостей, вбудованих в ядро фреймворка, існують пакети, що розширюють його можливості, Можливості, що надаються пакетами, а також повний перелік таких пакетів зручно відстежувати через спеціальний ресурс [7].

Веб-фреймворк Django використовується також в таких великих і відомих сайтах, як Instagram, Disqus, Mozilla, The Washington Times, Pinterest, YouTube, Google та інших [6, 8].

Метою даної роботи є дослідження особливостей роботи з базами даних у веб-фреймворку Django, яке включає налаштування під'єднання до бази даних, створення таблиць та записів у базі, створення моделей та створення і запровадження міграцій.

3. Обговорення особливостей роботи з базами даних у Django.

Щоб налаштувати з'єднання з базою даних, необхідно ввести зміни у файл `settings.py`. Даний файл представляє звичайний модуль `python` з глобальними змінними рівня модуля, що представляють налаштування Django. За замовчуванням в конфігурації використовується база даних SQLite, яку

включено в python і не потрібно інстальовати ніяких додаткових модулів. Якщо необхідно використовувати іншу базу даних, потрібно встановити відповідні прив'язки до неї та змінити наступні ключі в пункті «default» так, щоб вони відповідали налаштуванням з'єднання з базою даних:

- ENGINE – це двигок бази даних для її використання. Існують вбудовані бази даних, такі як "django.db.backends.sqlite3", "django.db.backends.postgresql", "django.db.backends.mysql" або "django.db.backends.oracle". На додаток до офіційно підтримуваних баз даних, існують так звані «бекенди», надані третіми сторонами, що дозволяють використовувати інші бази даних з Django, такі як SAP SQL Anywhere, IBM DB2, Microsoft SQL Server, Firebird, ODBC. Версії Django та функції ORM, що підтримуються цими неофіційними бекендами, значно, дещо відрізняються. Запити щодо конкретних можливостей цих неофіційних бекендів, а також будь-які запити підтримки повинні бути спрямовані на їх канали підтримки, надані кожною третьою стороною проекту.
- NAME – назва бази даних. Якщо використовується SQLite, база даних буде зберігатися у вигляді файлу. У цьому випадку NAME повинно містити повний абсолютний шлях, включаючи назву файлу. За замовчуванням приймається значення os.path.join (BASE_DIR, 'db.sqlite3'), яке зберігатиме файл у каталозі проекту.

Якщо в якості бази даних використовується не SQLite, а інша база, потрібно додати додаткові параметри, такі як USER, PASSWORD та HOST. Найпростіший можливий файл налаштувань для однієї бази даних за допомогою SQLite може мати наступний зміст:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': 'mydatabase',
    }
}
```

Приклад для налаштування з'єднання з базою даних PostgreSQL може мати наступний вигляд:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'mydatabase',
        'USER': 'mydatabaseuser',
        'PASSWORD': 'mypassword',
        'HOST': '127.0.0.1',
        'PORT': '5432',
    }
}
```

Після налаштування з'єднання з базою даних необхідно створити моделі: макет бази даних з додатковими метаданими. Модель являє собою єдине, остаточне джерело даних користувача. Вона містить основні поля та поведінку даних, які зберігаються. Django дотримується принципу DRY (Don't repeat yourself), тобто кожна окрема концепція та/або фрагмент даних повинні перебувати в одному, і тільки в одному місці. Надмірність погана. Нормалізація хороша. Отже, мета полягає в тому, щоб визначити модель даних в одному місці та автоматично отримувати від неї як можна більше необхідної корисної інформації. Цей принцип включає в себе міграції, які, на відміну від Ruby On

Rails, наприклад, повністю виведені з файла моделей. Вони, по суті, є лише історією, яку Django може прокручувати, щоб оновити схему бази даних для того, щоб вона відповідала поточним моделям.

Для прикладу зобразимо дві моделі [2, 8]: Question і Choice. Question містить власне запитання і дату його публікації. Модель Choice має два поля: текст вибору і поле підрахунку кількості голосів. Кожен Choice асоційований з відповідним Question. Ці концепти зображені у вигляді звичайних класів python:

```
from django.db import models

class Question(models.Model):
    question_text = models.CharField(max_length = 200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length = 200)
    votes = models.IntegerField(default = 0)
```

Програмний код є досить простим. Кожна модель представлена класом, який є підкласом `django.db.models.Model`. Кожна модель має ряд власних класових змінних, кожна з яких являє собою поле бази даних в моделі. Кожне поле представляється екземпляром класу `Field`. Наприклад, поле `CharField` призначене для символьних полів, а поле `DateTimeField` – для часових подій [8]. Ця інформація говорить Django, який тип даних утримує кожне поле. Ім'я кожного екземпляра поля (наприклад, `question_text` або `pub_date`) – це назва поля, у машино-дружньому форматі. Це значення буде використовуватися у коді Python, і база даних буде використовувати його як назву стовпця. Деякі поля потребують аргументів. Наприклад, `CharField` вимагає аргумент `max_length` – максимальної довжини символьного поля. Подібний підхід використовується не тільки в схемі бази даних, але й при валідації. Поле також може мати різні необов'язкові аргументи, а в даному випадку нами встановлено значення за замовчуванням 0. Зв'язки між моделями визначаються з використанням `ForeignKey`. Це говорить Django, що кожен вибір пов'язаний з одним питанням. Django підтримує всі можливі відносини з базою даних: "багато-до-одного", "багато-до-багатьох" та "один-до-одного" [8].

Перш ніж використовувати моделі, їх необхідно активувати. Для цього необхідно перш за все включити додаток до проекту. Щоб включити додаток у проект, потрібно додати посилання на його клас конфігурації в налаштуваннях `INSTALLED_APPS`. Після цього необхідно виконати команду `makemigrations`, в яку слід передати назву додатка. Під час запуску `makemigrations`, Django повідомляється, що було внесено деякі зміни до моделей і необхідно, щоб зміни збереглися як міграція. Міграції містять інформацію, як Django зберігає зміни до моделей, а, отже, і до схеми бази даних, і просто представляються файлами на фізичному носії. Існує команда `migrate`, яка стартує міграції і автоматично керує схемою бази даних. Дана команда виконує всі незастосовані міграції, а Django відстежує, які з них використовуються за допомогою спеціальної таблиці у базі даних з іменем `django_migrations` і запускає їх замість бази даних. Тобто, по суті, відбувається синхронізація змін, внесених до моделей, із схемою в базі даних. Міграції є дуже потужним інструментом, що дозволяють змінювати моделі протягом часу, коли проект розробляється без необхідності видаляти базу даних або таблиці та створювати нові. Міграції зазвичай спеціалізуються на оновленні бази даних в реальному часі без втрати даних [4, 8].

Отже, щоб ввести зміни до моделі, необхідно:

- Змінити моделі у файлі `models.py`;
- Виконати `python manage.py makemigrations` для створення міграції для цих змін;

- Виконати `python manage.py migrate`, щоб застосувати ці зміни до бази даних.

Причини того, що існують окремі команди для створення та застосування міграцій, пов'язані зі створенням чи модифікацією їх до своєї системи керування власними версіями та надсиланням їх власним додатком. Вони не тільки полегшують розробку, а і можуть бути використані іншими розробниками в ході виробничого програмування [8].

4. Висновки. У даній роботі було розглянуто веб-фреймворк Django для роботи з базами даних та, висвітлено особливості його архітектури, а також було з'ясовано особливості налаштування серверів для підтримки роботи даного веб-фреймворку. Проведені дослідження можливостей з'єднання з локальною та віддаленою базами даних PostgreSQL, MySQL, SQLite та Oracle виявили основні принципи та організацію роботи з моделями даних. Відмічається, що цей процес включає в себе створення моделі, її редагування, впровадження за допомогою міграцій, які, в свою чергу, можуть мати характерні особливості у їх створенні і застосуванні.

1. Jacob Kaplan-Moss, Adrian Holovaty. The Definitive Guide to Django: Web Development Done Right.
2. Nigel George. Build Your First Website with Python and Django.
3. Leif Azzopardi, David Maxwell, Tango With Django: A beginner's Guide to Web Development With Python / Django 1.9.
4. Daniel Roy Greenfield, Andrey Roy Greenfeld. Two Scoops of Django 1.11: Best Practices for the Django Web Framework.
5. www.djangopackages.com/
6. <https://uk.wikipedia.org/wiki/Django>.
7. <https://docs.djangoproject.com/en/2.0/>.

УДК 004.056

В.М. Мельник, К.В. Мельник, Б.В. Шульга

Луцький національний технічний університет

ПОРІВНЯННЯ ТРЬОХ НАЙПОПУЛЯРНІШИХ WEB-КАРКАСІВ ДЛЯ РОЗРОБКИ ПРОЕКТІВ В PYTHON

Мельник В.М., Мельник К.В., Шульга Б.В. Порівняння трьох найпопулярніших web-каркасів для розробки проектів в Python. В даній статті наведено результати досліджень особливостей застосування трьох найпопулярніших web-каркасів для розробки проектів у Python: Pyramid, Django та Flask. Відмічається, що найбільш гнучкий каркас Pyramid може бути застосований лише для створення невеликих додатків з відкритим кодом. Однак, Django надає чудовий набір компонентів за замовчуванням і з цієї причини він набуває популярності у виборі для створення середніх та великих веб-програм. Каркас Flask відмінно підходить для розробників, які працюють з невеликими проектами і потребують швидкого створення web-додатків, простого веб-інтерфейсу і вимагає невеликої конфігурації.

Всі три каркаси задовольняють відповідний список вимог, який змінює дизайн продукту та швидкість роботи нових функцій та виправлень. Тут Flask показує себе з найкращого боку і як Django може відчувати себе незграбним у невеликому проектному масштабі. Гнучкість і розширюваність Pyramid не відіграє важливої ролі для розробників, але слід також враховувати нові вимоги до динамічного технічного забезпечення, що постійно висуваються розробниками.

Мельник В.М., Мельник Е.В., Шульга Б.В. Сравнение трех самых популярных web-каркасов для разработки проектов в Python. В данной статье приведены результаты исследований особенностей применения трех самых популярных web-каркасов для разработки проектов в Python: Pyramid, Django и Flask. Отмечается, что наиболее гибкий каркас Pyramid может быть применен только для создания небольших приложений с открытым кодом. Однако, Django предоставляет замечательный набор компонентов по умолчанию и по этой причине он приобретает популярность в выборе для создания средних и крупных веб-приложений. Каркас Flask отлично подходит для разработчиков, работающих с небольшими проектами которые требуют быстрого создания web-приложений, простого веб-интерфейса и небольшой конфигурации.

Все три каркасы удовлетворяют соответствующий список требований, который меняет дизайн продукта и скорость работы новых функций и исправлений. Здесь Flask показывает себя с наилучшей стороны и как Django может чувствовать себя неуклюжим в небольшом проектном масштабе. Гибкость и расширяемость Pyramid не играет важную роль для разработчиков, но следует также учитывать и новые требования к динамическому техническому обеспечению, которые постоянно выдвигаются разработчиками.

Melnyk V.M., Melnyk K.V., Shulga B.V. A comparison of the three most popular web-frameworks for Python project development. This article presents the research results on the peculiarities of using the three most popular web-frames for Python project development: Pyramid, Django and Flask. It is noted that the most flexible framework Pyramid can only be used to create small applications with open source code. However, Django provides a wonderful set of components by default and for this reason it is becoming popular in choice for development of medium and large web applications. The Flask framework is great for developers who work with small projects and require the rapid creation of web-based applications, a simple web-interface and requires a small configuration.

All three frameworks meet the relevant requirements list, which changes product design and the speed of new features and fixes. Here Flask shows itself on the best side and how Django can feel awkward on a small projective scale. Flexibility and extensibility of Pyramid does not play an important role for developers, but it should also take into account the new requirements for dynamic technical support that are every time being put forward by developers.

1. Постановка проблеми. Світ веб-каркасів Python повний вибору: Django, Flask, Pyramid, Tornado, Bottle, Diesel, Pexen, Falcon та багато інших, що конкурують за інтересами розробників. Розробник може скоротити варіанти вибору і це допоможе завершити йому проект і перейти до наступного. В даній статті ми зосередимося на Flask, Pyramid та Django [5,9,12]. Ці каркаси охоплюють рішення від мікропроекту до веб-служби розміром підприємства.

Щоб допомогти зробити вибір між трьома простими (або, принаймні більш обґрунтованими) каркасами, ми можемо створити один і той же додаток у кожній структурі та порівняємо код, виділяючи сильні та слабкі сторони кожного підходу.

Flask являє собою "мікрокаркас", перш за все, спрямований на малі програми з більш простими вимогами. Pyramid і Django спрямовані на програми більшого об'єму, однак використовують вони різні підходи до розширюваності та гнучкості [7]. Pyramid – націлений на гнучкість і дозволяє розробнику використовувати потрібні інструменти для свого проекту. Це означає, що розробник може вибрати базу даних, структуру URL-адреси, стиль шаблону тощо. Django прагне включити всі "батареї" до веб-додатків, тому розробникам потрібно лише відкрити вікно і почати працювати, позичаючи багато модулів Django з коробки.

Django включає в себе ORM з коробки, а Pyramid та Flask залишають ці речі за розробником, щоб вибрати, як вони хочуть зберігати свої дані чи взагалі для збереження даних в цілому [1]. Найпопулярнішою програмою ORM для веб-додатків крім Django є SQLAlchemy [3,11]. Однак існує безліч інших варіантів від DynamoDB і MongoDB до простих локальних завдань, таких як LevelDB або простий SQLite. Все ж Pyramid призначений для використання будь-якого стійкого шару, навіть, можливо, ще не винайденого.

2. Аналіз останніх досліджень. Підхід з використанням "батарежок", включених в комплект Django дозволяє розробникам, які знають Python вже давно, занурюватися в веб-додатки швидко, без необхідності достроково приймати багато рішень про їх інфраструктуру. Django має власні інструменти для обробки шаблонів, форм, маршрутизації, автентифікації, адміністрування баз даних та інші. На відміну від Pyramid, який включає в себе маршрутизацію та автентифікацію, адміністрування шаблонів і баз даних вимагає зовнішніх бібліотек.

Вибір компонентів для додатків Flask та Pyramid дає більшу гнучкість для розробників, використання яких не відповідає стандартному ORM і потребує взаємодії з різними робочими процесами або шаблонами.

Flask – наймолодший з трьох каркасів, з'явився в середині 2010 року. Pyramid почав своє життя в проєкті Pylons і отримав назву Pyramid. Все ж перший реліз був здійснений у 2005 році. Django випустив свій перший реліз у 2006 році, незабаром після початку проєкту Pylons (Pyramid). Pyramid і Django – це надзвичайно досконалі каркаси, які мають багато плагінів та розширень для задоволення неймовірно великої кількості потреб.

Хоча Flask має більш коротку історію, але перехід від каркасів, що існували раніше, не буде великою складністю. Він міцно встановив свої пам'ятки на малих проєктах. Найчастіше використовується він у невеликих проєктах з лише однією або двома функціями. Один з таких проєктів – це `httpbin`, простий, але надзвичайно потужний помічник для налагодження та тестування бібліотек HTTP.

Що ж спільного в них? Найбільш активною на StackOverflow є спільнота Django з 165873 питань, зафіксованих в Інтернет-дослідженні та здоровим набором блогів від розробників та користувачів. Спільнота Flask та Pyramid не настільки великі, але їх спільноти досить активні у своїх списках розсилки та в IRC. Зібравши лише 20723 питань StackOverflow, Flask є у 8 разів менший, ніж Django. На Github вони мають майже однакову кількість зірок: 32580 – Django, 34029 – Flask. Всі три каркаси доступні під ліцензіями BSD.

Метою даної статті послужило дослідження порівняння трьох найпопулярніших web-каркасів для розробки проєктів в python. Зокрема, ставилося питання гнучкості у використанні, наповнення зручним інтерфейсом для розробників, використання ресурсів та інших програмних частин в ході розробки вищезгаданих додатків.

3. Основна частина. Що стосується завантаження, то Django та Pyramid поставляються з вбудованими інструментами для створення масивних програм-додатків. Flask – не містить нічого подібного, оскільки цільова аудиторія Flask не намагається створювати великі MVC-проєкти.

Сама найпростіша програма "Hello World!" є базовою, і багато IDE, які дозволяють вибрати тип проєкту, створюють щось подібне в якості шаблону. Наведемо простий приклад "Hello World!" з <http://flask.pocoo.org/tutorial/>:

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return "Hello World!"
if __name__ == '__main__':
    app.run()
```

Ось чому для Flask немає інструментів завантаження – на них не існує попиту. Це видно звичайному програмісту з наведеного вище прикладу Hello World, розміщеного на домашній сторінці Flask.

Для проектів, що потребують більшої відокремленості компонентів, Flask має у своєму функціоналі макети (blueprint). Наприклад, розробник може структурувати додаток Flask з усіма функціями, пов'язаними з користувачем, у `users.py` та функціями, пов'язаними з продажами, в `ecommerce.py`, а потім імпортувати їх і додавати їх у свій додаток на `site.py`.

Pyramid – це інструмент завантаження піраміди, що називається `pcreate` і є частиною Pyramid. Раніше набір інструментів "Вставити" передбачав завантаження, але потім це було замінено спеціальним інструментом для Pyramid, наприклад:

```
$ pcreate -s starter hello_pyramid
```

Пірамід, порівняно з Flask, призначений для великих і складних додатків. Через це його інструмент завантаження створює великий скелет проекту. Він також містить основні файли конфігурації. Приклад шаблону дерева та файли для додавання програми розробника для завантаження в індекс-пакета Python наведемо нижче.

```
hello_pyramid
├── CHANGES.txt
├── development.ini
├── MANIFEST.in
├── production.ini
├── hello_pyramid
│   ├── __init__.py
│   ├── статичний
│   │   ├── піраміди-16x16.png
│   │   ├── пірамід.png
│   │   ├── theme.css
│   │   └── theme.min.css
│   ├── шаблони
│   │   └── mytemplate.pt
│   ├── tests.py
│   └── views.py
├── README.txt
└── setup.py
```

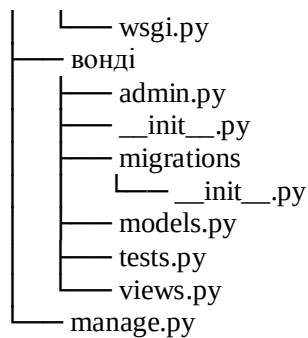
Як і в інших каркасах, завантажувач Pyramid є неймовірно гнучким. Він не обмежується однією програмою за замовчуванням; `pcreate` може використовувати будь-яку кількість шаблонів проектів. Включений у створення шаблон "starter" можна використовувати разом із проектами, які підтримують SQLAlchemy і ZODB. На PyPi можна знайти готові розроблені додатки для Google App Engine, jQuery Mobile, Jinja2, сучасних frontend фреймворків та багато іншого [2,10].

Django також має свій власний інструмент завантаження, побудований як частина `django-admin`.

```
django-admin startproject hello_django
django-admin startapp howdy
```

Ми вже можемо бачити один із способів, який відрізняє Django від Pyramid. Видно, що Django розділяє проект на індивідуальні програми, тоді як Pyramid і Flask очікують, що проект буде "єдиним додатком" з кількома видами (функції відображення `views`) або моделями. Тут можна відтворити структуру проекту на Flask та Pyramid, але це рішення не існує за замовчуванням.

```
hello_django
├── hello_django
│   ├── __init__.py
│   ├── settings.py
│   └── urls.py
```



За замовчуванням Django включає лише порожні моделі та файли шаблонів, так що новий користувач бачить трохи менше коду для прикладу. Це також (на жаль) залишає вибір за розробником, як розповсюджувати свій проект.

Недоліком цього інструменту і його завантаження є складність для розуміння новачкам. Якщо розробник раніше не упакував додаток, то процедура розгортання буде здаватись їм затратною в часі справою. Проекти з великою спільнотою, наприклад, *django-oscar*, упаковані та доступні на PyPi, але менші проекти на Github часто не мають єдиної упаковки.

І ще одна перспектива роботи з шаблонізаторами. Слід зазначити, що запуск додатка Python, який може обробляти HTTP-запити, є чудовим початком, але більшість активних користувачів не будуть зацікавлені в використанні Curl для взаємодії з веб-програмою розробника. На щастя, всі три суперники надають простий спосіб заповнити HTML-коди з користувацькою інформацією, а також дозволяють людям насолоджуватися розробленим чудовим інтерфейсом Bootstrap.

Шаблон дозволяє вставляти динамічну інформацію безпосередньо на сторінку без використання запитів AJAX. Це приємно з точки зору досвіду роботи з користувачем, оскільки розробнику потрібно лише зробити один запит, щоб отримати всю сторінку та всі її динамічні дані. Це особливо важливо на мобільних сайтах, де запити можуть тривати кілька секунд.

Всі параметри шаблонів, які можна побачити, покладаються на "контекст", який забезпечує динамічну інформацію для шаблону, в ході перетворення в HTML. Найпростіший спосіб використання шаблону полягає в тому, щоб заповнити ім'я входу в систему і привітати користувача належним чином. Можна було б використовувати AJAX, щоб отримати подібну динамічну інформацію, але для повного виклику, щоб просто заповнити ім'я користувача, даний підхід буде трохи надмірним у порівнянні з шаблонами.

Що стосується Django, то в прикладному використанні він дуже простий. Припустимо, що у нас є об'єкт користувача, який має властивість повного імені, який також містить ім'я користувача. У Python ми передали поточному користувачеві шаблон наступним чином [3,4]:

```
def a_view(request):
    return render_to_response(
        "view.html",
        {"user": cur_user})
```

Заповнення контексту шаблону настільки ж просте, як передавання словника об'єктів Python та структур даних, які повинен використовувати шаблон. Тепер нам потрібно відображати назву на сторінці, лише якщо вони забудуть, хто вони.

```
<!-- view.html -->
<div class="top-bar row">
  <div class="col-md-10">
    <!-- more top bar things go here -->
  </div>
  {% if user %}
  <div class="col-md-2 whoami">
    You are logged in as {{ user.fullname }}
```



```
</div>
{% endif %}
</div>
```

По-перше, ви помітите використання `{% if user %}`. У шаблонах Django `{%}` використовується для керуючих тверджень, таких як цикли та умовні оператори. Якщо команда `'if user'` використовується, щоб запобігти випадкам, коли немає користувача. Анонімні користувачі не повинні бачити залогінення на сайті.

У блоці `if` можна побачити, що включення імені є настільки ж простим, як обертання властивості, яку потрібно вставити в `{{ }}`. `"{{ "` – використовується для вставки фактичних значень у шаблон, наприклад `{{ user.fullname }}`.

Ще одним загальним використанням шаблонів є відображення груп предметів, таких як сторінка інвентаризації для комерційного сайту.

```
def browse_shop(request):
    return render_to_response(
        "browse.html",
        {"inventory": all_items})
```

У шаблоні ми також можемо використовувати оператор `"{%"`, щоб обійти цикл по всіх елементах інвентаря та заповнити URL-адресу своєї окремої сторінки.

```
{% for widget in inventory %}
<li><a href="/widget/{{ widget.slug }}">{{ widget.displayname }}</a></li>
{% endfor %}
```

Для виконання найбільш поширених завдань з шаблонізації Django може досягти мети, використовуючи лише декілька конструкцій, що і значно полегшує початок роботи.

Flask за замовчуванням використовує шаблонізатор Jinja2, але може бути налаштованим на використання іншого шаблонізатора. Обидва приклади DjangoTemplates, згадані вище, працюють і у Jinja2. Замість того, щоб перебирати ті ж приклади, можна розглянути ті місця, які відрізняють Jinja2 від DjangoTemplates.

Як Jinja2, так і DjangoTemplates надають функцію, яка називається фільтром, де список може бути переданий через функцію перед тим, як відобразитись [10]. Блог, в якому доступні категорії публікацій, можуть використовувати фільтри [1], щоб відображати їх категорії у списках, розділених комами.

```
<!-- DjangoTemplates -->
<div class="categories">Categories: {{ post.categories|join:", " }}</div>

<!-- Jinja2 -->
<div class="categories">Categories: {{ post.categories|join(", ") }}</div>
```

У шаблонізаторі Jinja можна передавати у фільтр будь-яку кількість аргументів, оскільки Jinja розглядає його як виклик функції Python в круглих дужках навколо аргументів. DjangoTemplates використовує двокрапку як роздільник між іменем фільтра та аргументом фільтра, який обмежує кількість аргументів до одного.

Виклик циклів виглядає подібним чином як у Jinja так і в DjangoTemplates. Подивимося, чим вони відрізняються. У Jinja2 конструкція `for-else-endfor` дозволяє розробнику прокручувати список, а також керувати випадком, коли немає елементів.

```
{% for item in inventory %}
<div class="display-item">{{ item.render() }}</div>
{% else %}
<div class="display-warn">
```

```
<h3>No items found</h3>
<p>Try another search, maybe?</p>
</div>
{% endfor %}
```

Версія Django цієї функції однакова, але використовує `for-empty-endfor`, а не `for-else-endfor`.

```
{% for item in inventory %}
<div class="display-item">{{ item.render }}</div>
{% empty %}
<div class="display-warn">
<h3>No items found</h3>
<p>Try another search, maybe?</p>
</div>
{% endfor %}
```

Окрім синтаксичних розбіжностей показаних вище, Jinja2 забезпечує більший контроль над середовищем виконання та розширеними функціями. Наприклад, можна відключити потенційно небезпечні функції, щоб безпечно виконувати ненадійні шаблони, або скомпілювати шаблони заздалегідь, щоб забезпечити їх дійсність (протидія ін'єкціям).

Як Flask, так і *Pyramid* підтримує багато шаблонізаторів, у тому числі Jinja2 і Mako, але за замовчуванням Pyramid використовує Chameleon – реалізацію шаблону ZPT (Zope Page Template). Давайте розглянемо наведений перший приклад, додавши ім'я користувача до верхньої панелі розробленого сайту. Код на Python виглядає так само, крім того, нам не потрібно явно називати функцію `render_template` [14].

```
@view_config(renderer='templates/home.pt')
def my_view(request):
    # TODO something
    return {'user': user}
```

Розроблений власний шаблон виглядає інакше. ZPT – це шаблонний стандарт на основі XML, тому ми використовуємо оператори, подібні до XSLT, для обробки даних.

```
<div class="top-bar row">
  <div class="col-md-10">
    <!-- more top bar things go here -->
  </div>
  <div tal:condition="user"
    tal:content="string:You are logged in as ${user.fullname}"
    class="col-md-2 whoami">
  </div>
</div>
```

Chameleon насправді має три різні області простору імен для дій шаблонів. TAL (Template Attribute Language) забезпечує такі основні інструменти, як умовні оператори, засоби форматування рядків та заповнення вмісту тегів [15]. Наведений вище приклад використовував TAL (Template Attribute Language) для завершення своєї роботи. Для більш складних завдань необхідні TALEX та METAL. TALEX (Template Attribute Language Expression Syntax) містить такі вирази, як розширене форматування рядків, оцінка виразів Python та імпорт виразів і шаблонів.

Однак METAL (Macro Expansion Template Attribute Language) є найпотужнішою (і складною) частиною шаблону Chameleon. Дані макроси легко розширюються, і їх можна визначити як слоти, які заповнюються при виклику макросу.

4. Висновки і перспективи. З проведеного дослідження можна зробити висновок, що Pyramid є найбільш гнучким каркасом з трьох, які бралися на увагу для дослідження. Він може бути використаний для невеликих додатків, а також в складних проєктах, наприклад – Dropbox. Спільності з відкритим кодом, такі як Fedora, вибирають його для таких програм, як система значків спільноти, яка отримує інформацію про події з багатьох інструментів проєкту, щоб призначати користувачам значки у стилі досягнення. Одна з найбільш поширених скарг на Pyramid полягає в тому, що він містить багато варіантів, що можуть завадити почати новий проєкт.

Однак, найбільш популярним каркасом є Django, і список сайтів, які використовують його – вражаючий. Bitbucket, Pinterest, Instagram та The Onion використовують Django для всіх або частини своїх сайтів. Для сайтів, із загальними потребами Django надає чудовий набір компонентів за замовчуванням і з цієї причини він став популярним вибором для створення середніх та великих веб-програм.

Каркас Flask відмінно підходить для розробників, які працюють з невеликими проєктами, що потребують швидкого створення простого веб-сайту з підтримкою Python. Це дозволяє завантажувати невеликі одноразові інструменти або прості веб-інтерфейси, побудовані за існуючими API. Вбудовані проєкти, які потребують простого веб-інтерфейсу, що швидко розвивається і вимагає невеликої конфігурації, часто користуються Flask на зовнішній панелі. Наприклад, jtvviewer, який надає веб-інтерфейс для перевірки журналів компіляторів PyPy в режимі реального часу.

Всі три каркаси змогли задовільнити невеликий список вимог, і можна побачити, чим вони відрізняються. Ці розбіжності не просто косметичні, вони також змінюють дизайн продукту та швидкість відправлення нових функцій та виправлень. Оскільки наведений приклад невеликий, то видно, де Flask показує себе з найкращого боку і як Django може відчувати себе незграбним у невеликому масштабі. Гнучкість і розширюваність Pyramid не стала важливим фактором, але слід враховувати, що в реальному світі постійно висуваються нові вимоги до "динамічного технічного забезпечення".

1. Мігель Грінберг – Flask. Створення Web-додатків, 2006, - 272 с.
2. Джефф Форс'є – Розробка Web лататків на Django, 2009, - 456 с.
3. Адріан Головатий - Django. Детальний довідник, 2010, - 550 с.
4. Володимир Дронов. Django: практика створення Web-сайтів, 2016, - 528 с.
5. <https://github.com/Pylons/pyramid>
6. <https://github.com/miguelgrinberg/Flask-SocketIO>
7. <https://www.djangoproject.com/>
8. <https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world-legacy>
9. <https://github.com/django/django>
10. <http://jinja.pocoo.org/>
11. <http://docs.sqlalchemy.org/en/latest/>

УДК 004.93

Мельник К.В., Мельник В.М., Мацібора А.С.
Луцький національний технічний університет

ДОСЛІДЖЕННЯ КОНТЕНТНИХ МЕТОДІВ РОЗПІЗНАВАННЯ СПАМУ

Мельник К.В., Мельник В.М., Мацібора А.С. Дослідження контентних методів розпізнавання спаму. В роботі надана загальна постановка методів розпізнавання спаму. Наведено огляд існуючих підходів до вирішення задачі розпізнавання спаму. Описані основні підходи, що використовуються в задачі розпізнавання спаму, визначено етапи процесу розпізнавання та розглянуті найбільш поширені математичні методи пошуку спаму. Розкрито особливості використання, переваги та недоліки зазначених методів. Зроблено висновок щодо необхідності подальшого розроблення алгоритмів розпізнавання на основі зазначених методів, що були б простими в реалізації, ефективними, мали низькі обчислювальні витрати при навчанні та високу якість класифікації в реальних завданнях.

Ключові слова: розпізнавання спаму, метод опорних векторів, штучні нейронні мережі.

Мельник К.В., Мельник В.М., Мацібора А.С. Исследование контентных методов распознавания спама. В работе представлена общая постановка методов распознавания спама. Приведен обзор существующих подходов к решению задачи распознавания спама. Описаны основные подходы, используемые в задаче распознавания спама, определены этапы процесса распознавания и рассмотрены наиболее распространенные математические методы поиска спама. Раскрыты особенности использования, преимущества и недостатки указанных методов. Сделан вывод о необходимости дальнейшей разработки алгоритмов распознавания на основе указанных методов, которые были бы простыми в реализации, эффективными, имели низкие вычислительные затраты при обучении и высокое качество классификации в реальных задачах.

Ключевые слова: распознавания спама, метод опорных векторов, искусственные нейронные сети.

Melnyk K.V., Melnyk V.M., Matsibora A.S. Exploration of content methods for spam detection. In this paper is given general resolution of the methods for spam detection. The existing methods review is given to solve the problem of spam detection. There are described some main approaches used for the task of spam recognition in this work with the steps determination on the recognition process and the most shared mathematical methods of spam searching. There are also revealed the using features with advantages and disadvantages of specified methods. It is made some conclusion about the further development necessity for recognition algorithms on the basis of these submitted methods that would be simply used in implementation, effective, with having low computing costs in the training process and high classification quality in the real tasks.

Keywords: spam detection, reference vector method, artificial neural network.

Актуальність та постановка проблеми. Спам у електронній пошті залишається проблемою, за даними [1], близько 60% всіх електронних листів є спамом. Хоча кількість небажаної кореспонденції зменшилась, але вона стала більш шкідливою. За допомогою спам листів поширюють не тільки рекламу, а й шкідливе програмне забезпечення, програми для вимагання грошей такі як Petya, Cryakl і Shade.

За останні роки було винайдено чимало способів боротьби з небажаною кореспонденцією. Нажаль, зловмисники стежать за протидією поширенню спаму й винаходять все нові прийоми для обходу фільтрів. Таким чином, всі дослідження в області боротьби з не запитуваною кореспонденцією надзвичайно актуальні в цей час.

Аналіз останніх досліджень та публікацій. Аналіз, розробку методів розпізнавання спаму досліджено в роботах А.М. Мироненка, П. Грехама, Викас П. Дешпанде, І. Терейковський [2–5]. Кожен із методів має свої переваги та недоліки, для практичного застосування необхідно виконати їх порівняння за функціональністю, сферою застосування, ефективністю тощо.

Мета дослідження – огляд основних методів розпізнавання спаму в тексті, розкрити їхні особливості використання, переваги та недоліки.

Виклад основного матеріалу дослідження та обґрунтування отриманих результатів. Спам — масова розсилка кореспонденції рекламного чи іншого характеру людям, які не висловили бажання її одержувати. Передусім термін «спам» стосується рекламних електронних листів.

Існує два основних підходи до боротьби з небажаною кореспонденцією, а саме визначення відправника як спамера, або аналізу листа, на основі чого робиться висновок, що він є спамом. Найбільш ефективним є комплексних захист (рис. 1) [6], що складається з наступних етапів: аналіз відправника; використання фільтрів; аналіз змісту листа.

Для аналізу змісту в листах застосовуються різноманітні методи, кожен з яких має свої переваги і особливості використання. Переважна більшість методів розпізнавання спаму так чи інакше засновані на припущенні, що текст, який відносяться до спаму, має однакові ознаки (слова чи

словосполучення), і наявність чи відсутність таких ознак в тексті визначає його приналежність чи неприналежність до тієї чи іншої категорії.

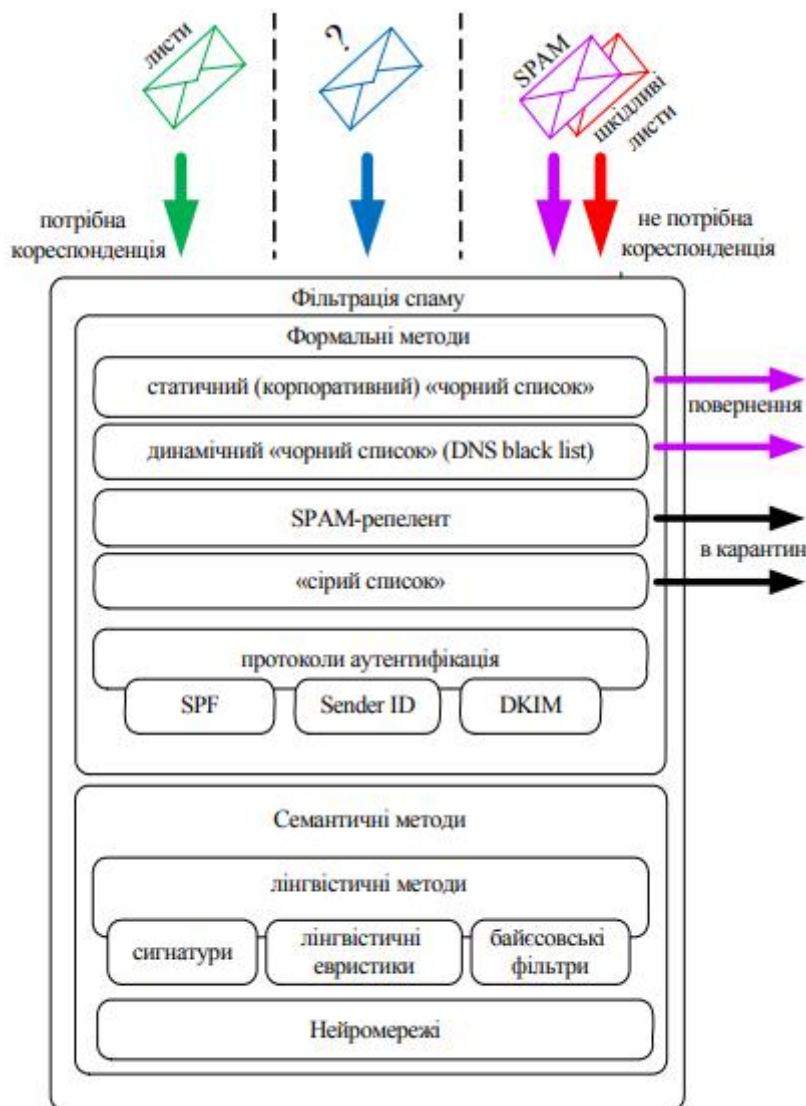


Рисунок 1. – Комплексний захист від спаму

Таким чином, задача фільтрації спаму, розглядається як задача класифікації – визначення належності об'єкта (електронного повідомлення) до одного з заздалегідь виділених класів (спам і «не спам») на підставі аналізу сукупності ознак, що характеризують даний об'єкт.

В даній статті описано та досліджено такі методи контекстного розпізнавання спаму в електронній пошті :

- Байєсівська (наївна) класифікація спаму;
- класифікація методом опорних векторів;
- розпізнавання за допомогою штучних нейронних мереж.

Для порівняння даних методів використовувалась база електронних листів розміщена на сайті UCI Machine Learning Repository [7].

З усієї електронної кореспонденції було виділено 57 ознак, що характеризують дані повідомлення, а 58 змінна визначає приналежність листа до класу спам чи «не спам».

Дана база налічує 4601 спостереження з яких 1813 (39,4%) – це спам, а 2788 (60,6%) – звичайні листи.

Теорема Байєса лежить в основі багатьох сучасних систем штучного інтелекту, призначених для роботи в умовах невизначеності. Такі системи дають ймовірнісну оцінку, тому звичайно не

заміняють експерта, а забезпечують підтримку прийняття рішення. Таким чином, для спаму повинна бути визначена множина ознак.

У загальному вигляді визначення найбільш вірогідного класу алгоритмом наївної байєсівської класифікації виглядає наступним чином:

Нехай $F_S(W_i)$ – кількість спам-листів, у яких зустрілося слово (W_i) , а $F_{NS}(W_i)$ – кількість корисних листів, у яких зустрілося слово W_i ; H_S – гіпотеза про те, що лист є спамом, H_{NS} – корисний лист. Тоді ймовірність того, що поява слова W_i у листі означає спам, обчислюється за формулою:

$$P(W_i|H_S) = \frac{F_S(W_i)}{F_S(W_i) + F_{NS}(W_i)},$$

а ймовірність того, що слово W_i не вказує на спам у листі:

$$P(W_i|H_{NS}) = \frac{F_{NS}(W_i)}{F_S(W_i) + F_{NS}(W_i)},$$

Якщо вектор W включає всі m слів нового листа, то ймовірність того, що він спам, обчислюється за формулою Байєса таким чином:

$$P(H_S|W) = \frac{\prod_{j=1}^m P(w_j|H_S)}{\prod_{j=1}^m P(w_j|H_S) + \prod_{j=1}^m P(w_j|H_{NS})}.$$

Віднесення листа до спаму або корисних листів виконується з врахуванням заданого програмістом, адміністратором, користувачем поштової програми спам-фільтрації значення ймовірності, яке становить 0,6–0,8.

В основі фільтра лежить список ознак, за якими проводиться аналіз повідомлення і обчислюється умовна ймовірність спамності за кожного ознакою. Загальна ймовірність спаму повідомлення визначається методом об'єднання всіх ймовірностей за теоремою Байєса.

Провівши класифікацію даної бази листів за алгоритмом наївної байєсівської класифікації було складено звіт таб. 1. З результатів дослідження ми бачимо, що даний спосіб помилково фільтрує 767 (16,67 % від усіх спостережень) звичайних листів, як спам і дає всього 81,4% правильної фільтрації повідомлень.

Таблиця 1. Звіт розпізнавання спаму алгоритмом наївної байєсівської класифікації

Назва класу	Загалом	Правильно	Помилково	Правильно %	Помилково %
«Не спам»	2110	2021	89	95,78	4,22
Спам	2491	1724	767	69,21	30,79
Всього	4601	3745	856	81,4	18,6

Основні переваги наївного байєсівського класифікатора простота реалізації і низькі обчислювальні витрати при навчанні та класифікації. У тих рідкісних випадках, коли ознаки дійсно незалежні (або майже незалежні), наївний байєсівський класифікатор (майже) оптимальний. Основним недоліком методу є відносно невисока якість класифікації в більшості реальних завдань. Зазначений метод часто використовується в якості базового методу при порівнянні різних методів машинного навчання.

Метод опорних векторів (Support Vector Machine, SVM) використовує процес пошуку площини вирішення, яка може розділити позитивні і негативні приклади в багатовимірному просторі функції, в якому навчальні документи представлені як вектори. Цей метод розроблений В. Вапником в 1995 році, був вперше застосований до задачі класифікації текстів Торстеном Джохімсом. У своєму первинному вигляді алгоритм вирішував завдання розрізнення об'єктів двох класів. Метод набув величезну популярність завдяки своїй високій ефективності. Багато дослідників використовують його в роботах, присвячених класифікації текстів. Підхід, запропонований В. Вапником для визначення того, до якого з двох заздалегідь визначених класів повинен належати аналізований зразок, заснований на принципі структурної мінімізації ризику.

Результати класифікації текстів за допомогою методу опорних векторів є одними з найкращих, у порівнянні з іншими методами машинного навчання. Однак, швидкість навчання даного алгоритму одна з найнижчих.

Провівши класифікацію даної бази листів за алгоритмом методом опорних векторів було складено звіт таб. 2. З результатів дослідження ми бачимо, що даний спосіб дає 89,18% правильної фільтрації повідомлень. Даний метод працює краще ніж алгоритм наївної байєсівської класифікації.

Таблиця 2. Звіт розпізнавання спаму методом опорних векторів

Назва класу	Загалом	Правильно	Помилково	Правильно %	Помилково %
«Не спам»	2788	2645	143	94,87	5,13
Спам	1813	1458	355	80,42	19,58
Всього	4601	4103	498	89,18	10,82

Штучні нейронні мережі (Artificial Neural Network). Штучні нейронні мережі набули широкого вивчення в галузі аналізу даних з 1986 року. Вони представляють собою математичну модель, а також її програмні або апаратні реалізації, побудовані за подібністю мереж нервових клітин живого організму. Нейронні мережі - це один з найбільш відомих і старих методів машинного навчання.

Штучні нейронні мережі - це адаптивна система, яка складається з групи з'єднаних штучних нейронів. Система може бути навчена для зміни її внутрішніх станів, відображення зв'язків повідомлень та їх класів. Для ефективного проведення фільтрації листів необхідно визначити раціональну структуру і топологію нейронної мережі. Основні топології класифікуючих нейронних мереж - це одно- і багатошаровий персептрон, мережа радіально базисних функцій. Всі вищевказані топології мають високу точність в обробці одночасно лінійних і нелінійних прикладів, але прийняття рішень щодо класифікації важко формалізуються у зв'язку з природою організації нейронної мережі та представляють нетривіальну задачу з урахуванням масштабованості з обмеженими обчислювальними ресурсами.

Для дослідження було побудовано декілька різних нейронних мереж таб. 3.

Таблиця 3. Опис нейронних мереж

№	Архітектура	Продуктивність	Алгоритм	Функція помилки	Активация прихованих нейронів	Активация вихідних нейронів
1	MLP 57-19-2	96,43	BFGS 175	Сума квадратів	Експонента	Тотожна
2	MLP 57-24-2	95,34	BFGS 75	Крос-ентропія	Експонента	Софтмак
3	MLP 57-8-2	94,47	BFGS 53	Крос-ентропія	Експонента	Софтмак
4	MLP 57-8-2	96,21	BFGS 71	Сума квадратів	Гіперболічна	Логістична
5	RBF 57-30-2	60,53	RBFT	Сума квадратів	Гауссіан	Тотожна

Найкращі результати з них показали нейронні мережі № 1, № 4 та № 2.

Таблиця 4. Звіт розпізнавання спаму MLP 57-19-2 алгоритм BFGS 175

Назва класу	Загалом	Правильно	Помилково	Правильно %	Помилково %
«Не спам»	2788	2697	91	96,74	3,26
Спам	1813	1711	102	94,37	5,63
Всього	4601	4408	193	95,81	4,19

Таблиця 5. Звіт розпізнавання спаму MLP 57-8-2 алгоритм BFGS 71

Назва класу	Загалом	Правильно	Помилково	Правильно %	Помилково %
«Не спам»	2788	2716	72	97,42	2,58
Спам	1813	1690	123	93,27	6,73
Всього	4601	4406	195	95,78	4,23

Таблиця 6. Звіт розпізнавання спаму MLP 57-24-2 алгоритм BFGS 75

Назва класу	Загалом	Правильно	Помилково	Правильно %	Помилково %
«Не спам»	2788	2688	100	96,41	3,59
Спам	1813	1684	129	92,89	7,12
Всього	4601	4372	229	95,02	4,98

Отже, ми бачимо що нейроні мережі дають найкращі результати, тому це перспективний напрямок подальшого дослідження. Поєднання двох різних архітектур нейронних мереж може дати дуже продуктивні результати.

Висновки та перспективи подальшого дослідження. В результаті виконаного дослідження було проаналізовано переваги та недоліки таких методів розпізнавання спаму в електронних листах: Байєсівська (наївна) класифікація спаму, класифікація методом опорних векторів та розпізнавання за допомогою штучних нейронних мереж. Було побудовано декілька різних нейронних мереж та порівняно їхню продуктивність з іншими способами фільтрації електронної кореспонденції.

В майбутньому дані проведеного дослідження можна використати для розробки спам-фільтра для електронної скриньки кафедри. Перспективним є дослідження продуктивності різних топологій нейронних мереж для виконання даного завдання.

1. traffic from January 2014 to September 2017, by month [Електронний ресурс] — Режим доступу: <https://www.statista.com/statistics/420391/spam-email-traffic-share/>
2. Vikas P. Deshpande. An Evaluation of Naive Bayesian Anti-Spam Filtering Techniques / Vikas P. Deshpande, Robert F. Erbacher, Chris Harris // Proceedings of the 2007 IEEE Workshop on Information Assurance United States Military Academy, West Point, 2007. — NY 20—22 June. — Режим доступу: <http://digital.cs.usu.edu/~erbacher/publications/Bayes-Vikas2.pdf>.
3. Graham P. A Plan for Spam / P. Graham, 2002. [Електронний ресурс] — Режим доступу: <http://www.paulgraham.com/spam.html>.
4. Мироненко А. Н. Алгоритм контентной фильтрации спама на базе совмещения метода опорных векторов и нейронных сетей : автореф. дис. на соискание науч. степени канд. техн. наук: спец. 05.13.19 «Методы и системы защиты информации, информационная безопасность» / А. Н. Мироненко. — СПб., 2012. — 18 с.
5. Терейковський І. Методологія класифікації листів електронної пошти з використанням нейронних мереж. / І. Терейковський // Захист інформації – 2013, – Том 15, №2, – С. 115-122.
6. Кузьма К., Зівенко В. Аналіз методів фільтрації електронної пошти від спаму. / К. Кузьма, В. Зівенко // Науковий журнал – Геометричне моделювання та інформаційні технології № 1 (3), квітень 2017. – Миколаїв : МНУ імені В. О. Сухомлинського, 2017. — С. 84-89.
7. Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.

УДК 004.415.3

Міскевич О.І., Сичов Д.І., Христинець А.О.
Луцький національний технічний університет

ПРО МОДЕРНІЗАЦІЮ ЛОКАЛЬНО-ОБЧИСЛЮВАЛЬНОЇ МЕРЕЖІ НА ПрАТ «ВОЛИНЬОБЛЕНЕРГО» НА ОСНОВІ GRE-TUNNEL З ВИКОРИСТАННЯМ ШИФРУВАННЯ IPSEC

Міскевич О.І., Сичов Д.І., Христинець А.О. Про модернізацію локально-обчислювальної мережі на ПрАТ «Волиньобленерго» на основі GRE-tunnel з використанням шифрування IPsec. Створено нову мережу та модернізовано вже існуючу локально-обчислювальну мережу на основі обладнання MikroTik.

Ключові слова: мережа, локально-обчислювальна мережа, MikroTik, IPsec, комутатор, модернізація.

Міскевич О.И., Сычев Д.И., Христинец А.О. О модернизации локально-вычислительной сети на ЧАО «Волиньоблэнерго» на основании GRE-tunnel с использованием шифрование IPsec. Создано новую сеть и модернизировано уже существующую локально-вычислительную сеть используя оборудование MikroTik.

Ключевые слова: сеть, локально-вычислительная сеть, MikroTik, IPsec, коммутатор, модернизация.

Miskevych O. I., Sychov D. I., Khrystinets A.O. On modernization of the local-computer network at PrAT "Volynoblenergo" on the basis of the GRE-tunnel using IPsec encryption. New network has been created and an existing locally-based computer network based on MikroTik equipment has been upgraded.

Keywords: network, local-computer network, MikroTik, IPsec, switch, modernization.

Вступ. На сьогодні у світі існує безліч комп'ютерів і понад 80% із них об'єднані в різні інформаційно-обчислювальні мережі, від малих локальних мереж у офісах, до глобальних мереж, типу Internet. ПрАТ «Волиньобленерго» - це підприємство, основними видами діяльності якого є передача електричної енергії місцевими (локальними) мережами та постачання електроенергії за регульованим тарифом. Енергопостачання споживачів забезпечується лініями та обладнанням із напругою 0,4-110кВ.

Постановка проблеми. ПрАТ «Волиньобленерго» обслуговує понад 353 тис. фізичних та 10 тисяч юридичних споживачів. У зв'язку зі створенням нового структурного підрозділу, виникла необхідність у розробці нового вирішення питання організації інформаційно-обчислювальної мережі на базі вже існуючого комп'ютерного парку та програмного комплексу, який відповідає сучасним науково-технічним вимогам та з урахуванням зростаючих потреб і з можливістю подальшого поступового розвитку мережі (у зв'язку з появою нових технічних і програмних рішень). Тому було вирішено модернізувати вже існуючу мережу, шляхом переходу на нове обладнання MikroTik із вбудованим програмним забезпеченням та створити нову локально-обчислювальну мережу.

Аналіз існуючих рішень. На ПрАТ «Волиньобленерго» використовується топологія «Зірка». В цьому випадку кожен комп'ютер підключається окремим кабелем до загального пристрою, який називають концентратором і який знаходиться в центрі мережі. У функції концентратора входить прийом і передача комп'ютером інформації одному або решті комп'ютерів із мережі. Головна перевага цієї топології перед загальною шиною - це істотно велика надійність. Будь-які неприємності з кабелем стосуються лише того комп'ютера до якого цей кабель приєднаний і лише несправність концентратора може вивести з ладу всю мережу. Крім того, концентратор може грати роль інтелектуального фільтру інформації, яка поступає від вузлів в мережу і, при необхідності, блокує заборонені адміністратором передачі. Проте, існує ряд недоліків, зокрема, вища вартість мережевого устаткування через необхідність придбання концентратора. Крім того, можливості із накопичення кількості вузлів в мережі обмежуються кількістю портів концентратора. Іноді має сенс будувати мережу з використанням декількох концентраторів ієрархічно сполучених між собою зв'язками типу зірка. В даний час ієрархічна зірка є найпоширенішим типом топології зв'язків як в локальних так і глобальних мережах. На рисунку 1 зображена детальна будова нинішньої мережі на одній із філій ПрАТ «Волиньобленерго».

У власній модернізації було використано обладнання компанії MikroTik, зокрема комутатори CRS125-24G-1S-in, тому що вони відповідають багатьом критеріям: досить низька ціна [2], вони

мають 24 порти Gigabyte Ethernet та володіють прекрасною пропускною спроможністю. На цих комутаторах встановлена операційна система Mikrotik RouterOS L 5, через яку зручно налаштовувати повністю весь комутатор під себе [4]. З проведеного аналізу MikroTik CRS125-24G-1S-RM виявився не просто комутатором, а 24-портовим маршрутизатором з необмеженим функціоналом [3]. Крім того, слід зауважити, що таке обладнання використовують у провідних інтернет-компаніях світу [1].

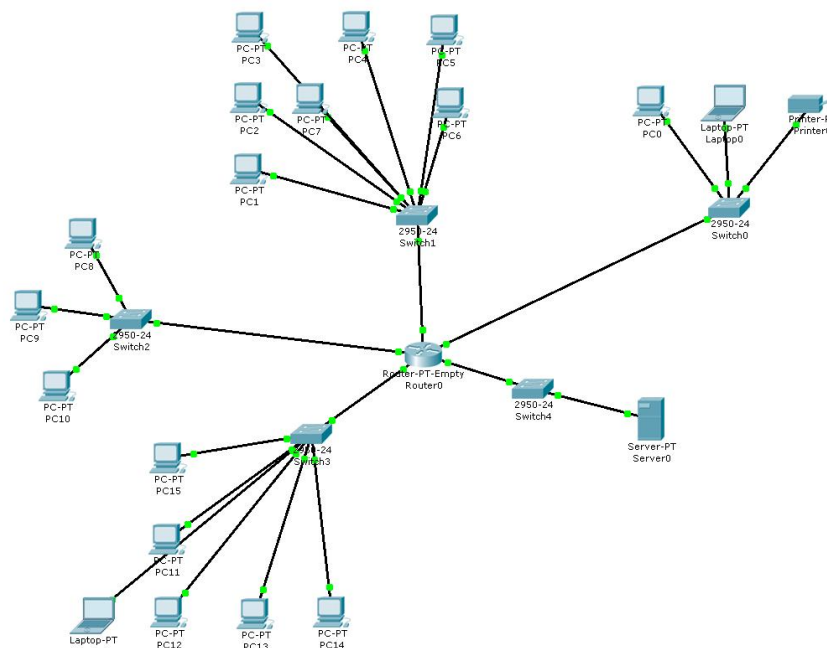


Рисунок 1 - Схема мережевого обладнання на ПрАТ “Волиньобленерго”

Виклад основного матеріалу роботи Вся маршрутизація мережі побудована на обладнанні MikroTik CRS125-24G-1S- in та на протоколі OSPF. Потрібно зазначити, що протокол OSPF (Open Shortest Path First) належить до класу протоколів стану каналів (Link State Protocol). Дослівний переклад — першим обирається найкоротший шлях. «Open»-специфікація протоколу вільно поширюється, на відміну, наприклад, від специфікації протоколу EIGRP. RFC 2328 — основний діючий документ по OSPF. Окрім того, протокол OSPF дозволяє визначити для будь-якої мережі значення метрики, залежно від типу послуги TOS (Type of Service). В OSPF підтримуються метрики пропускної здатності та затримки. Метрика, що оцінює пропускну здатність каналу, визначається, наприклад, компанією Cisco, як кількість секунд, необхідних для передачі 100 Мбіт. Метрика затримки — час у мікросекундах, необхідний маршрутизатору для обробки, постановки в чергу та передачі пакетів. Для кожної з метрик протокол OSPF будує окрему таблицю маршрутизації. Стандартний порядок розрахунку метрики, що оцінює показники надійності, затримки й вартості, поки не визначений. Цей порядок визначається адміністратором.

У цьому протоколі також закладено можливість балансування навантаження на шляхах як з однаковою вартістю, так і з різною. Розподіл трафіку відбувається пропорційно метриці шляху.

Generic Routing Encapsulation (GRE-tunnel) - простий протокол тунелювання. Це означає, що можна взяти початкові дані разом зі службовими заголовками (як правило, це IP, але може бути і Ethernet і ATM), запакувати в пакет і передати до публічної мережі, немов машина їде в тунелі через гори. На кінцевому вузлі заголовки нового пакету знімаються, а ваші дані в початковому вигляді продовжують свою подорож. GRE-тунелі є односпрямованими, і зазвичай мається на увазі наявність зворотного тунелю на іншій стороні, хоча взагалі кажучи, це необов'язково. Але в нашому випадку, коли посередині Інтернет, і завдання - організувати приватну мережу, зі зворотного боку повинне бути симетричне налаштування.

IPsec (скорочення від IP Security) — набір протоколів для забезпечення захисту даних, що передаються за допомогою протоколу IP, дозволяє здійснювати підтвердження справжності та/або шифрування IP-пакетів. IPsec також містить в собі протоколи для захищеного обміну ключами в мережі Інтернет.

Протоколи IPsec, на відміну від інших добре відомих протоколів SSL та TLS, працюють на мережевому рівні (рівень 3 моделі OSI). Це робить IPsec гнучкішим, так що він може використовуватися для захисту будь-яких протоколів, що базуються на TCP та UDP. IPsec може використовуватися для забезпечення безпеки між двома IP-вузлами, між двома шлюзами безпеки або між IP-вузлом і шлюзом безпеки. Протокол є "надбудовою" над IP-протоколом, і обробляє сформовані IP-пакети. IPsec може забезпечувати цілісність та / або конфіденційність даних переданих по мережі [6].

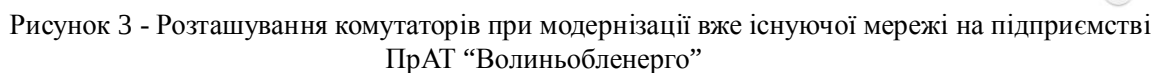
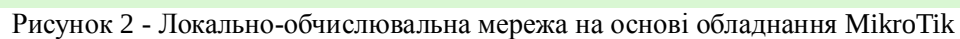
Зробимо порівняльну характеристику нинішнього обладнання структурного підрозділу ПрАТ "Волиньобленерго" та обладнання, яке було встановлене до модернізації. Результати порівняння подані у таблиці 1.

Таблиця 1. Порівняльна характеристика обладнання D-Link DES-1210 та MIKROTIK CRS125-24G-1S-IN

Назва	D-Link DES-1210	MIKROTIK CRS125-24G-1S-IN
Тип	Комутатор керований рівня 2	Комутатор керований
Кількість портів Fast Ethernet (10/100)	24 (з грозозахистом)	1 (консоль)
Кількість портів Gigabit Ethernet (10/100/1000)	2x SFP + 2x комбо 1000Base-T / SFP (з грозозахистом)	24
Інші порти	консольний RJ-45	1x SFP, 1x USB, 1x micro-USB
Моніторинг та конфігурування	Web-інтерфейс, SNMP, RMON, Telnet, SSH, LLDP	Mikrotik RouterOS L 5
Живлення	100-240 В, 50-60 Гц	100-240 В, 50 / 60Гц
Розміри, мм	440x140x44	246x135x50
Особливості	802.1Q, Q-in-Q, ISM VLAN, 802.1p, IGMP Snooping, STP, LBD, LACP, ACL, 802.1x, IMPB, DHCP Snooping	Налаштування портів для комутації або для маршрутизації. LCD сенсорний екран.

Легко бачити, що обладнання із правого стовбця є на порядок кращим та має свою ОС, в якій можна розвинути власну мережу, включити та налаштувати власний VPN із використанням GRE-tunnel та шифрування каналу з використанням протоколу IPSec без втрати швидкості передачі даних. Також, є можливість розгорнути декілька мереж із використанням одночасно декількох провайдерів із надійними налаштуваннями безпеки. На завершення, слід зауважити, що цією системою можна керувати не виходячи з дому.

Після проведеної модернізації локально-обчислювальна мережа на структурному підрозділі ПрАТ "Волиньобленерго" матиме наступний вигляд (рисунки 2):



1. <https://weblance.com.ua/336-amazon-ispolzuet-v-svoih-datacentrah-i-oblachnyh-servisah-oborudovanie-mikrotik.html>
2. <https://rozetka.com.ua/routers/c80193/producer=cisco-sb,mikrotik/>
3. <https://mikrotik.com/aboutus>
4. <https://habrahabr.ru/post/265387/>
5. <https://habrahabr.ru/post/271707/>
6. <https://uk.wikipedia.org/wiki/IPsec>

УДК 004.415.2

Муляр В. П., к. пед. н., доцент

Східноєвропейський національний університет імені Лесі Українки

ОСНОВИ РОЗРОБКИ ДОДАТКІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ JAVAFX

Муляр В. П. Основи розробки додатків з використанням технології JavaFX. У статті розглянуто особливості розробки графічного інтерфейсу користувача з використанням технології JavaFX. Розкрито структуру JavaFX-додатків. Розглянуто використання лямбда-виразів для обробки подій. Висвітлено питання компоновання елементів керування графічного інтерфейсу з використанням мови розмітки FXML та середовища SceneBuilder. Розглянуто особливості зміни зовнішнього вигляду інтерфейсу користувача за допомогою таблиць стилів CSS. Особливу увагу приділено анімації і трансформації зображень в JavaFX. Доведено високу ефективність технології JavaFX для розробки додатків із насиченим графічним інтерфейсом, які розгортаються на різних платформах.

Ключові слова: JavaFX, графічний інтерфейс користувача, компоновання, мова розмітки FXML, таблиці стилів CSS, анімація, трансформація.

Муляр В. П. Основы разработки приложений с использованием технологии JavaFX. В статье рассмотрены особенности разработки графического интерфейса пользователя с использованием технологии JavaFX. Раскрыта структура JavaFX-приложений. Рассмотрено использование лямбда-выражений для обработки событий. Освещены вопросы компоновки элементов управления графического интерфейса с использованием языка разметки FXML и среды Scene Builder. Рассмотрены особенности изменения внешнего вида интерфейса с помощью таблиц стилей CSS. Особое внимание уделено анимации и трансформации изображений в JavaFX. Доказана высокая эффективность технологии JavaFX для разработки приложений с насыщенным графическим интерфейсом, которые разворачиваются на разных платформах.

Ключевые слова: JavaFX, графический интерфейс пользователя, компоновки, язык разметки FXML, таблицы стилей CSS, анимация, трансформация.

Muliar V. P. Fundamentals of development of applications with use JavaFX technology. The article discusses the features of developing a graphical user interface using JavaFX technology. The structure of JavaFX applications is revealed. The use of lambda expressions for processing events is considered. Issues related to the layout of GUIs using the FXML markup language and the Scene Builder environment. The features of changing the appearance of the interface using CSS style sheets are considered. Particular attention is paid to animation and image transformation in JavaFX. The high efficiency of JavaFX technology is proven for developing rich graphical user-friendly applications, which unfold on different platforms.

Keywords: JavaFX, graphical user interface, layouts, FXML markup language, CSS style sheets, animation, transformation.

Постановка проблеми в загальному вигляді. Потужним інструментом для розробки настільних і мережевих додатків є технологія JavaFX, що дозволяє розробникам проектувати, створювати, тестувати, налагоджувати і розгортати насичені клієнтські додатки, які працюють для різних платформ. Програмний код JavaFX-додатка може посилатися на Application Programming Interface (API) будь-якої бібліотеки Java.

Починаючи з JavaFX 2.2, всі пізніші версії повністю інтегровані з Java SE Runtime Environment (JRE) і комплектом Java Development Kit (JDK). Оскільки JDK доступний для всіх основних настільних платформ (Windows, Mac OS X і Linux), то JavaFX-додатки, скомпільовані в JDK, можуть працювати на даних платформах. Крос-платформна сумісність дає змогу використовувати досвід узгодженого виконання додатків для JavaFX-розробників і користувачів. Платформа JavaFX призначена для забезпечення додатків такими складними функціями графічного інтерфейсу користувача, як плавна анімація, веб-перегляд, відтворення аудіо та відео, стилі на основі каскадних таблиць стилів CSS [2, с. 386].

Платформа JavaFX забезпечує графічні й анімаційні можливості для додатків, які дозволяють створювати і відображати зображення за допомогою програмного забезпечення і комп'ютерного устаткування. Компоненти графічного інтерфейсу користувача (Graphical User Interface, GUI) JavaFX-додатка утворюють сцену, логічна структура якої описується графом сцени. Відображенням GUI-інтерфейсу JavaFX-додатка є графічне зображення графа сцени. Для відображення GUI-інтерфейсу, розробленого на основі платформи JavaFX, і створення графічного зображення графа сцени, середовище виконання JavaFX Runtime представляє графічну систему, яка містить відповідні модулі і створює набір паралельних потоків. Основний потік JavaFX-додатка JavaFX Application Thread відповідає за оновлення сцени, обробку анімації і подій. Потік рендеринга системи Prism відповідає за відрисовку сцени, а фоновий медіапотік відповідає за декодування, буферизацію і відтворення аудіо та відео. Синхронізація графа сцени з його графічним представленням здійснюється за допомогою подій Pulse, які генеруються середовищем виконання JavaFX Runtime з максимальною частотою 1/60 секунди і посилаються в чергу подій

під час анімації і кожен раз, коли змінюється граф сцени, викликаючи перерисовку сцени з використанням компонування і стилів [3, с. 14].

Аналіз досліджень і публікацій. Розробці RIA-додатків (Rich Internet Applications) із використанням технології JavaFX присвячено дослідження Ю. Парфенова та В. Федорченко [4]. Розгляду архітектури платформи JavaFX 2.0, її основним компонентам графічного інтерфейсу користувача, використанню CSS-стилів, створенню візуальних ефектів, трансформації й анімації зображень, використанню компонентів JavaFX NetBeans, мові FXML присвячена робота Т. Машніна [3]. Аналізу технологій розробки насичених інтернет-додатків на платформі Java присвячено дослідження В. Герасимова та В. Левицької [1]. Особливості побудови графічного контенту додатків із використанням JavaFX і Swing компонентів і даних, взятих із баз даних, розкрито в дослідженні В. Карашецького [2]. Більшість дослідників вважають, що у Java-розробників з'явився потужний конкурентоспроможний інструмент – JavaFX. Технологія має багато різних реалізованих елементів інтерфейсу та класів, які дуже прискорюють розробку RIA-додатків.

Формулювання цілей статті. Мета статті – розглянути основи розробки графічного інтерфейсу користувача засобами JavaFX.

Виклад основного матеріалу. JavaFX – це базовий набір інструментів для розробки інтерфейсу користувача, єдине узгоджене середовище програмування додатків як для вбудованих, так і для настільних систем [10]. Іншими словами, JavaFX має на меті використання в багатьох типах пристроїв, таких як мобільні пристрої, смартфони, телевізори, планшетні комп'ютери та десктопи [6]. Для компіляції і запуску JavaFX-програм більше немає необхідності у встановленні додаткових програм. Усі частини одного проекту тепер можна реалізувати на одній платформі Java, що, в свою чергу, забезпечує швидкодію, підвищує безпеку, скорочує час розробки і економить витрати на розробку.

Розглянемо основні можливості JavaFX як засобу розробки графічного інтерфейсу користувача.

Структура додатків JavaFX. Загалом, JavaFX-додаток має три основні компоненти, а саме: підмостки, сцену та вузли [8]. Усе, що потрібно відобразити засобами JavaFX, розміщується на сцені, яку можна оформити і оживити «акторами», в ролі яких виступають елементи керування і форми. Сцена повинна знаходитися на підмостках, які є вікном верхнього рівня, якщо програма виконується на робочому столі операційної системи, або прямокутною областю, якщо програма виконується в вигляді аплета. Підмостки передаються у вигляді параметра методу start(), який необхідно перевизначити в підкласі, похідному від класу Application, як показано в наведеному нижче прикладі.

```
public class HelloWorld extends Application {  
    public void start(Stage stage) {  
        Label message = new Label("Hello, JavaFX!");  
        message.setFont(new Font(100));  
        stage.setScene(new Scene(message));  
        stage.setTitle("Hello");  
        stage.show();  
    }  
}  
  
public class MyApp extends Application {  
    public static void main(String[] args) {  
        launch(args);  
    }  
    ...  
}
```

Обробка подій. Керування графічними інтерфейсами здійснюється за допомогою подій. Коли користувач вибирає кнопки клацанням на них, регулює повзунки і виконують інші дії, графічний інтерфейс користувача реагує і оновлюється. Окремий елемент управління (наприклад, кнопка) забезпечується обробником подій, щоб отримувати повідомлення про вибір кнопки клацанням на ній. Все це істотно спрощується завдяки використанню лямбда-виразів, як показано в наведеному нижче прикладі коду.

```
red.setOnAction(event -> message.setTextFill(Color.RED));
```

При виборі кнопки клацанням на ній викликається лямбда-вираз. У даному випадку задається червоний колір тексту.

Властивості JavaFX. Властивість є атрибутом класу, значення якого можна читати або записувати. Як правило, властивість підтримується полем, а метод отримання і установки просто читає і записує дані у властивість відповідно. Властивість в JavaFX, крім методів отримання і установки, забезпечується третім методом, що повертає об'єкт класу, що реалізовує інтерфейс Property. Наприклад, у властивості text в JavaFX є метод Property <String> textProperty(). До об'єкту властивості можна приєднати приймач подій. Цим JavaFX відрізняється від колишньої технології JavaBeans. У JavaFX об'єкт властивості, а не компонент JavaBeans, посилає повідомлення про зміни. І для таких змін є вагомі підстави. Для реалізації прив'язаних властивостей в JavaBeans був потрібний шаблонний код, який виконував введення, видалення і запуск приймачів подій. А в JavaFX все зроблено набагато простіше, оскільки все навантаження беруть на себе бібліотечні класи [5, с. 86].

Прив'язка. Головне призначення властивостей в JavaFX полягає в повідомленні про прив'язку – автоматичному оновленні однієї властивості при зміні іншої. Механізм прив'язки дозволяє подолати труднощі, які виникають під час програмування інтерфейсу користувача. Як приклад розглянемо поле дати і селектор дати з календаря. Коли користувач вибирає дату з календаря, поле дати має бути оновлено автоматично, а разом із ним і властивість дати в моделі.

Компонування. Якщо графічний інтерфейс користувача містить кілька елементів управління, вони повинні бути розташовані на екрані як з точки зору виконуваних ними функцій, так і з точки зору зручності застосування. Компонування елементів управління графічного інтерфейсу користувача можна здійснити за допомогою спеціального інструментального засобу, який призначений для розробки інтерфейсу. З іншого боку, компоновка може бути досягнута програмним способом, тобто написанням коду в методі установки, що вводить елементи управління в призначений для користувача інтерфейс на окремих позиціях. Ще один спосіб полягає в тому, щоб визначити компонування на декларативній, непроцедурній мові. Наприклад, веб-сторінки компонуються засобами HTML і CSS. Аналогічно на платформі Android є окрема мова XML для визначення компоновок. У JavaFX підтримуються всі три згаданих вище способи. Зокрема, Scene Builder виконує в JavaFX роль візуального конструктора графічного інтерфейсу користувача [9].

Мова розмітки FXML. Для опису компонувань в JavaFX застосовується мова розмітки FXML. Файли FXML-документів можна створити вручну або ж скористатися для цієї мети інструментальним засобом Scene Builder. Отримавши такий файл, його можна завантажити в такий спосіб:

```
public void start(Stage stage) {  
    try {  
        Parent root = FXMLLoader.load(getClass().getResource("dialog.fxml"));  
        stage.setScene(new Scene(root));  
        stage.show();  
    } catch (IOException ex) {  
        ex.printStackTrace();  
        System.exit(0);  
    }  
}
```

Установити зв'язок між елементами управління інтерфейсу користувача і програмою можна, зокрема, скориставшись таким способом. Зокрема, анотацію @FXML можна використовувати для «впровадження» керуючих об'єктів у клас контролера. А в класі контролера повинен бути реалізований інтерфейс Initializable. У методі initialize() з класу контролера засоби прив'язки зв'язуються з обробниками подій. Як контролер можна використовувати будь-який клас – навіть сам JavaFX-додаток.

У файлі FXML-документа слід надати імена змінних екземпляра контролера для відповідних елементів управління в цьому файлі. Для цієї мети слугує атрибут fx:id. У кореневому елементі необхідно також оголосити клас контролера за допомогою атрибуту fx:controller. При завантаженні файлу FXML-документа конструється граф сцени, а посилання на іменовані об'єкти управління впроваджуються в ановані поля об'єкта контролера. Потім викликається його метод initialize(). Більшу частину ініціалізації можна зробити в файлі FXML-документа. Зокрема, можна визначити прості прив'язки і задати ановані методи контролера в якості

приймачів подій. Такий підхід дозволяє відокремити візуальний дизайн від поведінки програми. При цьому розробник призначеного для користувача інтерфейсу може займатися графічним оформленням програми, а програміст – реалізацією її поведінки.

Таблиці стилів CSS. У JavaFX допускається зміна зовнішнього вигляду інтерфейсу користувача за допомогою таблиць стилів CSS, що, як правило, зручніше, ніж надання атрибутів FXML-розмітки або виклик методів в Java. Таблицю стилів CSS можна завантажити програмно і застосувати її до графу сцени наступним чином:

```
Scene scene = new Scene(pane);  
scene.getStylesheets().add("scene.css");
```

У таблиці стилів можна звертатися до будь-яких елементів управління, які мають ідентифікатор. Як приклад розглянемо, як організувати управління зовнішнім виглядом панелі типу GridPane. Ідентифікатор цієї панелі встановлюється в коді наступним чином:

```
GridPane pane = new GridPane();  
pane.setId("pane");
```

А будь-яке заповнення проміжків або розстановка встановлюється не в коді, а в таблиці CSS, як показано нижче.

```
#pane {  
-fx-padding: 0.5em;  
-fx-hgap: 0.5em;  
-fx-vgap: 0.5em;  
-fx-background-image: url("metal.jpg")  
}
```

На жаль, замість відомих атрибутів CSS доведеться освоїти і застосовувати спеціальні атрибути JavaFX, імена яких починаються із префікса -fx- і утворюються з імен властивостей в нижньому регістрі і дефісів замість змішаного написання.

Замість стилізації за окремими ідентифікаторами можна скористатися класами стилів. Із цією метою клас стилю вводиться в вузловий об'єкт наступним чином:

```
HBox buttons = new HBox();  
buttons.getStyleClass().add("buttonrow");  
Потім він стилізується за допомогою наступного позначення класу CSS:  
.buttonrow {  
-fx-spacing: 0.5em;  
}
```

Крім того, таблиці стилів CSS можна застосовувати в FXML-компоновках. Для цього досить приєднати таблицю стилів до кореневої панелі наступним чином:

```
<GridPane id = "pane" stylesheets = "scene.css">
```

Код FXML-розмітки забезпечується атрибутами id або styleClass, як показано в наведеному нижче прикладі.

```
<HBox styleClass = "buttonrow">
```

Потім більшу частину стилізації можна вказати в таблиці стилів CSS, а FXML-розмітку використовувати тільки для перегляду [5, с. 102].

Анімація і трансформація. Основною ідеєю анімації в JavaFX є зміна властивості вузла протягом певного періоду часу. Якщо ця змінна визначає розташування вузла, результуюча анімація – це рух. Якщо з часом змінити властивості заповнення форми, ілюзія анімації змінює свою перспективу. Отже, анімація не завжди означає певну форму руху. Це може бути зміна кольору, прозорості тощо.

Платформа JavaFX забезпечує створення двох видів анімації – анімацію по ключовим кадрам і анімацію з вбудованою тимчасовою шкалою. JavaFX-анімацію представляє пакет javafx.animation, базовим класом якого є клас Animation. Клас Animation розширюється класами Timeline і Transition, при цьому клас Timeline представляє анімацію по ключовим кадрам, а клас Transition – анімацію з вбудованою тимчасовою шкалою.

Клас Animation має набір властивостей, що дозволяють управляти швидкістю і напрямком анімації, затримкою і кількістю циклів анімації, встановлювати автореверс анімації, зчитувати статус анімації, обробляти завершення анімації та ін.

Швидкість і напрям анімації можна встановити за допомогою метода `setRate(double value)`, затримку анімації – за допомогою метода `setDelay(Duration value)`, зчитати статус `Animation.Status.PAUSED`, `Animation.Status.RUNNING` або `Animation.Status.STOPPED` анімації –

методом `getStatus()`, встановити обробник завершення анімації – методом `setOnFinished(EventHandler<ActionEvent>value)`. Також клас `Animation` містить методи управління життєвим циклом анімації:

- `jumpTo(Duration time)` – перехід анімації до вказаної позиції на часовій шкалі;
- `playFrom(Duration time)` – запуск анімації, починаючи з вказаної позиції на часовій шкалі;
- `play()` – запуск анімації з поточної позиції на часовій шкалі;
- `playFromStart()` – запуск анімації з початкової позиції на часовій шкалі;
- `stop()` – зупинка анімації;
- `pause()` – пауза анімації [3, с. 247].

Анімація по ключовим кадрам дозволяє створити видимі зміни значення будь-якої JavaFX-властивості за певний проміжок часу за допомогою класу `Timeline`. Примірник класу `Timeline` можна створити за допомогою класу-фабрики `TimelineBuilder` або за допомогою одного з конструкторів, що дозволяють встановити частоту кадрів і набір ключових кадрів анімації:

```
public Timeline(double targetFramerate)
public Timeline(double targetFramerate, KeyFrame ... keyFrames)
public Timeline(KeyFrame ... keyFrames) і public Timeline()
```

Набір ключових кадрів `Timeline`-анімації можна поповнити методом `getKeyFrames().AddAll()`, а зупинити `Timeline`-анімацію і повернути її до початкової позиції – методом `stop()`. Ключовий кадр `Timeline`-анімації представлений класом `javafx.animation.KeyFrame` і визначає зміну значень JavaFX-властивостей за певний проміжок часу. Примірник класу `KeyFrame` можна створити за допомогою набору конструкторів, що дозволяють встановити час відтворення ключового кадру, ім'я ключового кадру, обробник закінчення ключового кадру і набір зміни значень відповідних JavaFX-властивостей:

```
public KeyFrame(Duration time, java.lang.String name, EventHandler<ActionEvent> onFinished,
java.util.Collection<KeyValue<?>> values)
public KeyFrame(Duration time, java.lang.String name, EventHandler<ActionEvent> onFinished,
KeyValue<?>... values)
public KeyFrame(Duration time, EventHandler<ActionEvent> onFinished, KeyValue<?>... values)
public KeyFrame(Duration time, java.lang.String name, KeyValue<?>... values)
public KeyFrame(Duration time, KeyValue<?>... values).
```

Зміна значення JavaFX-властивості представлена класом `javafx.animation.KeyValue`, екземпляр якого можна створити за допомогою конструкторів, дозволяє встановити змінювану JavaFX-властивість, її кінцеве значення в результаті анімації і спосіб її зміни протягом анімації:

```
public KeyValue(WritableValue<T> target, T endValue, Interpolator<? super T> interpolator)
public KeyValue(WritableValue<T> target, T endValue)
```

Спосіб зміни значення JavaFX-властивості протягом анімації представлений класом `javafx.animation.Interpolator`.

`Transition`-анімація з вбудованою часовою шкалою також використовує об'єкт `Interpolator` як значення властивості `interpolator` класу `Transition`.

`Transition`-анімація з вбудованою часовою шкалою, на відміну від `Timeline`-анімації, описує зміну в часі обмеженого набору JavaFX-властивостей, таких як прозорість (клас `FadeTransition`), положення в просторі (класи `PathTransition` і `TranslateTransition`), обертання (клас `RotateTransition`) і масштабування вузла графа сцени (клас `ScaleTransition`), а також колір заповнення (клас `FillTransition`) і колір контуру форми `Shape` (клас `StrokeTransition`). Іншими словами, `Transition`-анімація визначає цілий ряд монтажних переходів, що змінюють властивості вузлів у часі. Як приклад нижче показано, яким чином в коді реалізується збільшення розмірів вузла наполовину в напрямку обох осей координат x і y протягом трьох секунд.

```
ScaleTransition st = new ScaleTransition(Duration.millis(3000));
st.setByX(1.5);
st.setByY(1.5);
st.setNode(yesButton);
st.play();
```

Клас `PauseTransition` дозволяє зробити паузу в послідовності анімацій. Клас `PauseTransition` має властивість `duration` (тривалість паузи) і конструктори `public PauseTransition(Duration duration)` і `public PauseTransition()`, а також клас-фабрику `PauseTransitionBuilder`. Клас `AnimationTimer` дозволяє створювати таймер, що викликається в кожному кадрі анімації. Створити таймер можна, якщо розширивши абстрактний клас `AnimationTimer` і перевизначити його метод `handle(long`

now), що викликається в кожному кадрі. Для управління таймером клас AnimationTimer пропонує методи start() і stop().

Пакет javafx.scene.transform платформи JavaFX забезпечує трансформацію вузлів графа сцени, яка складається з афінних перетворень: обертання, переміщення, масштабування і зсуву. На відміну від анімації, трансформація графічних об'єктів не має плавного видимого переходу від початкової до кінцевої точки під час певного проміжку часу, а виконується відразу.

Базовим класом JavaFX трансформацій є клас javafx.scene.transform.Transform, який має реалізацію в вигляді класів Affine, Rotate, Scale, Shear і Translate. Застосувати JavaFX трансформації до вузла графа сцени можна двома способами. Перший спосіб – використати метод getTransforms() класу javafx.scene.Node, який повертає список ObservableList<Transform> об'єктів Transform, доповнити який можна методом addAll(). Інший спосіб – це застосування методів setRotate(), setRotationAxis(), setScaleX(), setScaleY(), setScaleZ(), setTranslateX(), setTranslateY(), setTranslateZ() класу javafx.scene.Node, який забезпечує трансформацію обертання, переміщення і масштабування для вузла графа сцени [3, с. 247].

Клас Affine представляє афінні перетворення матриці:

$$\begin{bmatrix} m_{xx} & m_{xy} & m_{xz} & t_x \\ m_{yx} & m_{yy} & m_{yz} & t_y \\ m_{zx} & m_{zy} & m_{zz} & t_z \end{bmatrix}$$

за допомогою властивостей: m_{xx} – X-множник матриці; m_{xy} – XY-множник матриці; m_{xz} – XZ-множник матриці; t_x – зсув по осі x; m_{yx} – YX-множник матриці; m_{yy} – Y-множник матриці; m_{yz} – YZ-множник матриці; t_y – зсув по осі y; m_{zx} – ZX-множник матриці; m_{zy} – ZY-множник матриці; m_{zz} – Z-множник матриці; t_z – зсув по осі z.

Екземпляр класу Affine можна створити за допомогою класу-фабрики AffineBuilder, за допомогою конструктора public Affine() або за допомогою статичного методу affine класу Transform, який повертає Affine-об'єкт. Афінні перетворення відображають n-вимірний об'єкт в n-вимірний, зберігають паралельність ліній і площин, а також пропорції паралельних об'єктів. За допомогою афінних перетворень можна створювати трансформації обертання, зсуву, масштабування і переміщення.

Для забезпечення обертання, масштабування, зсуву, переміщення вузла графа сцени використовуються класи Rotate, Scale, Shear та Translate відповідно.

Удосконалені елементи управління. JavaFX передбачає створення і використання комбінованих списків, панелей вкладок, дерев, таблиць, а також нові елементи управління графічного інтерфейсу, в тому числі селектор дат і меню-гармошку. Окрім того, засобами JavaFX можна побудувати багато видів діаграм. І для цього не потрібно спеціально встановлювати сторонні бібліотеки [7].

Розгортання додатків. JavaFX містить пакувальник додатків, який дозволяє створити простий у розгортанні автономний настільний додаток, що містить усі Java-бібліотеки часу виконання, необхідні для установки і запуску додатків.

Висновки. JavaFX є потужним інструментальним засобом для розробки графічного інтерфейсу користувача високої продуктивності, що включає аудіо, відео, графіку та анімацію. Для обробки подій використовуються лямбда-вирази, що істотно спрощує процес керування графічними інтерфейсами. Механізм прив'язки дозволяє реалізувати автоматичне оновлення однієї властивості при зміні іншої. Для побудови графічного інтерфейсу JavaFX пропонує використовувати декларативну мову розмітки FXML. Дизайнер може кодувати в FXML або використовувати JavaFX Scene Builder для інтерактивного проектування графічного інтерфейсу користувача. Scene Builder генерує розмітку FXML, яку можна перенести на IDE, де розробник може додати бізнес-логіку. JavaFX Scene Builder є ключовим засобом розробки для платформи JavaFX. Технологія JavaFX надає можливість змінювати зовнішній вигляд інтерфейсу користувача за допомогою таблиць стилів CSS, що, як правило, зручніше, ніж надання атрибутів FXML-розмітки або виклик методів в Java. Платформа JavaFX забезпечує створення двох видів анімації – анімацію по ключовим кадрам і анімацію з вбудованою часовою шкалою. Пакет javafx.scene.transform забезпечує трансформацію вузлів графа сцени, яка складається з афінних перетворень: обертання, переміщення, масштабування і зсуву. На відміну від анімації, трансформація графічних об'єктів не має плавного видимого переходу від початкової до кінцевої точки під час певного проміжку часу, а виконується відразу. JavaFX містить удосконалені елементи управління, які забезпечують побудову діаграм, опрацювання мультимедійних даних тощо.

Напрямом подальшого дослідження є розробка насичених мережових додатків, які розгортаються на різних платформах.

1. Герасимов В. В. Аналіз технологій розробки насичених інтернет-додатків на платформі Java / В. В. Герасимов, В. Я. Левицька // IT проектирования, моделирования, дизайна, WEB, 2017. – С. 355–363.
2. Карашецький В. П. Побудова графічного контенту додатків з використанням JavaFX і Swing компонентів і даних, взятих із баз даних / В. П. Карашецький. – Науковий вісник НЛТУ. – 2015. – Вип. 25.1. – С. 386–392.
3. Машнин Т. С. JavaFX 2.0: разработка RIA-приложений / Т. С. Машнин. – СПб.: БХВ-Петербург, 2012. – 320 с.
4. Парфенов Ю. Э. Разработка «насыщенных» интернет-приложений с помощью JavaFX / Ю. Э. Парфенов, В. Н. Федорченко // Системы обработки информации. – 2012. – Вип. 8 (106). – С. 40–46.
5. Хорстманн К. Java SE 8. Вводный курс / К. Хорстманн. – М.: Вильямс, 2014. – 208 с.
6. Carl Dea. JavaFX 2.0: Introduction by Example, apress, 2011. – 181 p.
7. Carl David Granbäck. Rich Internet Applications (RIAs): A Comparison Between Adobe Flex, JavaFX and Microsoft Silverlight. – Chalmers University of Technology – Göteborg, 2009. – P. 1–70.
8. JavaFX – Application [Електронний ресурс]. – Режим доступу: https://www.tutorialspoint.com/javafx/javafx_application.htm
9. JavaFX Scene Builder [Електронний ресурс]. – Режим доступу: <http://www.oracle.com/technetwork/java/javase/downloads/javafxscenebuilder-info-2157684.html>
10. Kishori Sharan. Learn JavaFX 8 (The Expert's Voice in Java): Building User Experience and Interfaces with Java 8. – New York, 2015. – P. 1–1173.

УДК 004.415.3

Пех П.А., Кобук Р.Ф.

Луцький національний технічний університет

С++BUILDER ПРОГРАМНО-ДЕМОНСТРАЦІЙНИЙ КОМПЛЕКС ДЛЯ ДОСЛІДЖЕННЯ РІВНЯНЬ З ПАРАМЕТРАМИ МЕТОДОМ ПЕРЕТИНІВ ПРЯМИМИ ЛІНІЯМИ

Пех П. А., Кобук Р. Ф. С++Builder програмно-демонстраційний комплекс для дослідження рівнянь з параметрами методом перетину прямими лініями. В статті запропоновано програмно-демонстраційний комплекс засобами середовища С++Builder для дослідження рівнянь з параметрами методом перетинів прямими лініями. Наведені також результати розв'язування кількох рівнянь.

Ключові слова: С++Builder проект, метод перетинів, перетин горизонтальними лініями, перетин бісекторними лініями, графічне розв'язування рівнянь.

Пех П. А., Кобук Р. Ф., С++Builder програмно-демонстрационный комплекс для исследования уравнений с параметрами методом сечений прямыми линиями. В статье предложено программно-демонстрационный комплекс средствами С++Builder для исследования уравнений с параметрами методом сечений прямыми линиями. Приведены также результаты решения нескольких уравнений.

Ключевые слова: С++Builder проект, мето сечений, сечение горизонтальными линиями, сечение бисекторными линиями, графическое решение уравнений.

Pekh Petro, Kobuk Roman. Software C++ Builder software=demonstration complex to study equations with parameters by the cross-section method. In the article the software-demonstration complex developed in C ++ Builder environment is offered for research of equations with parameters by straight lines cross-sections method. The results of solving several equations are also given.

Keywords: C ++ Builder project, the intersections method, the intersection of horizontal lines method, the cross section of the bisector lines method, the graafic solution of the equations.

Постановка задачі. Алгебраїчні та трансцендентні рівняння з параметрами досить поширені в науковій та технічній областях. Здебільшого з допомогою параметрів враховують особливості форми конструкцій, міцність матеріалів і тому подібні речі, в той час, як з допомогою змінних та функцій описують процеси, що досліджуються. Можна також стверджувати, що на теперішній час немає єдиної методики, як досліджувати рівняння з параметрами. Як правило, розв'язування рівняння з параметрами – це кожного разу оригінальне дослідження, кожне з яких потребує такого ж оригінального підходу. Зупинимось, однак, на деяких спільних рисах досліджень рівнянь з параметрами. Розв'язуючи рівняння з параметрами, необхідно:

- досліджувати область допустимих значень як невідомого $D(f)$, так і параметра $D(a)$ рівняння;
- досліджувати область зміни значень функції $f(x)$ і параметра a , враховувати всі можливі обмеження цих значень;
- забезпечувати рівносильність можливих перетворень рівняння;
- враховувати умови монотонності зміни значень функцій, що входять до складу рівняння.

Розглянемо кілька типових задач, у яких потрібно розв'язати алгебраїчне або трансцендентне рівняння з параметром, і покажемо, як зазначені вище підходи можуть бути використані у кожному з них.

Задача 1. Знайти всі значення параметра a , при кожному з яких рівняння

$$\sin(x + 4a) + \sin((x^2 - 6x - 7a)/2) = 4x - x^2 - a$$

не має дійсних розв'язків.

Якщо позначити $y = (x^2 - 6x - 7a)/2$, то рівняння прийме такий вигляд:

$$\sin(x + 4a) + \sin y = 4x - (2y + 6x + 7a) - a.$$

Або ж: $2y + \sin y = (-x - 4a) + \sin(-x - 4a).$

Далі вводимо у розгляд функцію $f(t) = 2t + \sin t$, яка є монотонно зростаючою на всій числовій осі, що дозволяє знайти розв'язок даного рівняння, записавши це рівняння у вигляді $f(y) = f(-x - 4a)$. Звідси: $y = -x - 4a$. Отже, тут ми використали допоміжну функцію $f(t)$ і факт її монотонного зростання на всій числовій осі.

Задача 2. Знайти всі значення параметра k , при кожному з яких рівняння

$$(1 + (2 - k)\sin t)/(\cos t - \sin t) = 2k$$

має хоча б один розв'язок на інтервалі $(0; \pi/2)$.

Очевидною є заміна $x = \cos t$; $y = \sin t$ і дослідження цього рівняння через нововведені змінні.

Задача 3. Дослідити розв'язки рівняння залежно від значень параметра a :

$$(a-3)x^2 - 2(3a-4)x + 7a-6 = 0.$$

Тут варто знайти дискримінант D , і з умови $D \geq 0$ визначити інтервал зміни параметра a , за якого рівняння має дійсні корені.

Задачі на дослідження рівнянь з параметрами можна умовно поділити на дві групи:

- задачі, у яких потрібно знайти кількість розв'язків залежно від значень параметра, або довести, що ці розв'язки не існують;
- задачі, у яких потрібно знайти розв'язки рівняння залежно від значень параметра.

Всі наведені вище підходи до розв'язування рівнянь з параметрами є традиційними, простіше кажучи, аналітичними [1]. Наша мета – пришвидшити процес розв'язку рівнянь за рахунок використання комп'ютерних технологій. Він базується на тому, що будь-яке рівняння можна подати у вигляді $f(x) = 0$. Тому розв'язок рівняння можна звести до побудови графіка функції $f(x)$ на певному діапазоні зміни значень аргументу за вибраних значень параметрів, що входять до складу рівняння, і визначення точок перетину графіка функції з віссю абсцис. Тому, поєднуючи аналітичні дослідження з паралельною побудовою за допомогою програми графіка досліджуваної функції, можна набагато швидше і ефективніше розв'язати рівняння з параметром.

Метою нашого дослідження є розроблення засобами C++Builder середовища програмно – демонстраційного комплексу для розв'язування рівнянь з параметрами типу $f(x) = \varphi(x, a)$ методом перетинів прямими лініями шляхом використання сучасних інформаційних технологій. Вибраний для досліджень тип рівнянь з параметрами відрізняється тим, що в лівій частині рівняння знаходиться функція $f(x)$, що не містить параметрів, а в правій частині – функція $\varphi(x, a)$ з параметром a . До того ж, як це витікає з назви статті, ця функція має своїм графіком сімейство прямих ліній. Точки перетину графіка функції $f(x)$ зі сімейством прямих ліній, які описуються функцією $f(x)$, є шуканими розв'язками рівняння.

Варто зазначити, що проблема розв'язування задач з параметрами привертала увагу багатьох дослідників, однак залучення для вирішення цієї проблеми новітніх інформаційних технологій розпочалося порівняно недавно. На нашу думку, пропонуваний нам підхід може бути рекомендований як для більш ефективного розв'язування зазначеного класу рівнянь, так і для потреб навчального процесу.

Основна частина. Планом передбачалося проведення досліджень у три етапи:

- вибір методу розв'язування рівняння;
- розроблення програмного комплексу засобами середовища C++Builder для розв'язування рівнянь з параметрами;
- перевірка адекватності отриманої моделі шляхом порівняння розв'язків з тими, що отримані аналітично.

Метод перетинів розв'язування рівнянь з параметрами.

Перетин функції $y = f(x)$ сімейством горизонтальних прямих ліній $\varphi(x, a) = a$.

Для кожного дійсного значення параметра a розв'язати рівняння [1]:

$$(x+1) \cdot |x-1| = a.$$

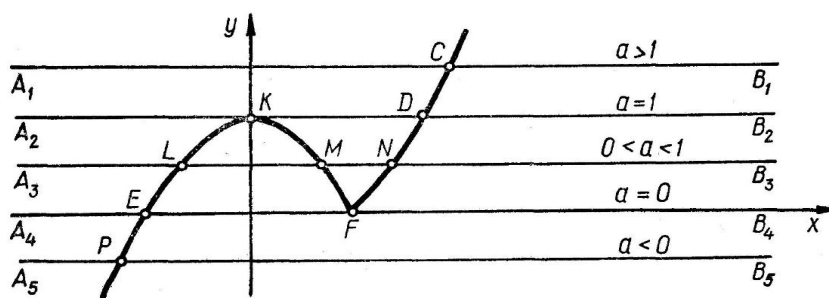


Рис.1 – Метод перетинів горизонтальними лініями

У даному випадку: $f(x) = y_1 = (x + 1) \cdot |x - 1|$ і $\varphi(x, a) = y_2 = a$.

Виходячи з означення абсолютної величини, функцію y_1 можна подати у вигляді:

$$y_1 = \begin{cases} x^2 - 1, & \text{якщо } x \geq 1, \\ 1 - x^2, & \text{якщо } x < 1. \end{cases}$$

Графіком функції y_1 є парабола, яка на проміжку $x \geq 1$ вітками направлена вгору і з вершиною в точці $(0; -1)$, а на проміжку $x < 1$ вітками направлена вниз і з вершиною в точці $(0; 1)$.

Функція $y_2 = a$ визначає сімейство горизонтальних прямих ліній, паралельних осі абсцис.

Залежно від значення параметра a , задане в умові рівняння може мати один, два або три розв'язки. Ці розв'язки відображаються на малюнку у вигляді точок перетину графіків функцій $f(x) = y_1 = (x + 1) \cdot |x - 1|$ та $\varphi(x, a) = y_2 = a$.

Загалом задане в умові рівняння рівносильне сукупності двох рівнянь:

$$\begin{cases} x^2 - 1 = a, & \text{якщо } x \geq 1, \\ 1 - x^2 = a, & \text{якщо } x < 1. \end{cases}$$

Якщо $a < 0$ (пряма A_5B_5), то заданому рівнянню задовільняє тільки один (менший) корінь другого рівняння сукупності, тобто $x_1 = -\sqrt{1-a}$ (точка P).

Якщо $a = 0$ (пряма A_4B_4), то заданому рівнянню задовільняє два корені відповідного квадратного рівняння: спільний корінь обох рівнянь сукупності $x_1 = 1$ (точка F), і корінь другого рівняння сукупності $x_2 = -1$ (точка E).

Якщо $0 < a < 1$ (пряма A_3B_3), то заданому рівнянню задовільняють три корені відповідного квадратного рівняння: більший корінь першого рівняння сукупності $x_1 = \sqrt{a+1}$ (точка N) і обидва корені другого рівняння сукупності $x_2 = -\sqrt{1-a}$ (точка L) та $x_3 = \sqrt{1-a}$ (точка M).

Якщо $a = 1$ (пряма A_2B_2), то заданому рівнянню задовільняють два корені відповідного квадратного рівняння: більший корінь першого рівняння сукупності $x_1 = \sqrt{2}$ (точка D) і корінь другого рівняння сукупності $x_2 = 0$ (точка K).

Якщо $a > 1$ (пряма A_1B_1), то заданому рівнянню задовільняє тільки один (більший) корінь першого рівняння сукупності, тобто $x_1 = \sqrt{a+1}$ (точка C).

Перетин функції $y = f(x)$ сімейством прямих ліній - бісектрис $\varphi(x, a) = a$.

Для кожного дійсного значення параметра a розв'язати рівняння [1]:

$$|x - 1| + |x + 1| = x - a.$$

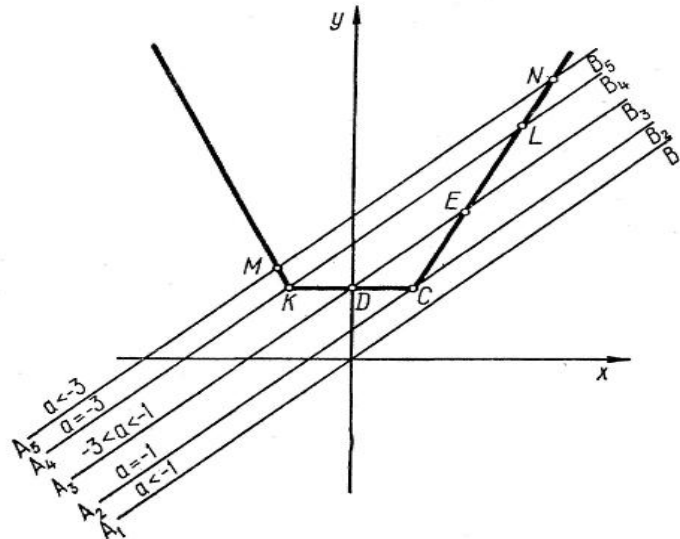


Рис.2 – Метод перетинів лініями-бісектрисами

У даному випадку: $f(x) = y_1 = |x - 1| + |x + 1|$ і $\varphi(x, a) = y_2 = x - a$.

Виходячи з означення абсолютної величини, функцію y_1 можна подати у вигляді:

$$y_1 = \begin{cases} -2x, & \text{якщо } x < -1; \\ 2, & \text{якщо } -1 \leq x < 1; \\ 2x, & \text{якщо } x \geq 1; \end{cases}$$

Функція $y_2 = x - a$ визначає сімейство прямих ліній, паралельних прямій $y = x$, яка є бісектрисою першого та третього координатних кутів – звідси і назва методу.

Загалом задане в умові рівняння рівносильне сукупності таких рівнянь:

$$\begin{cases} 2x = a - x, & \text{якщо } x < -1; \\ 2 = x - a, & \text{якщо } -1 \leq x < 1; \\ 2x = x - a, & \text{якщо } x \geq 1. \end{cases}$$

Якщо $a > -1$ (пряма A_1B_1), то задане рівняння не має розв'язків.

Якщо $a = -1$ (пряма A_2B_2), то заданому рівнянню задовільняє тільки один спільний корінь другого і третього рівнянь сукупності, тобто $x_1 = 1$ (точка C).

Якщо $-3 < a < -1$ (пряма A_3B_3), то заданому рівнянню задовільняють два корені: один корінь другого рівняння сукупності, тобто $x_1 = 2 + a$ (точка D), і корінь третього рівняння сукупності, тобто $x_2 = -a$ (точка E).

Якщо $a = -3$ (пряма A_4B_4), то заданому рівнянню задовільняють два корені: один корінь є спільним для другого та третього рівняння сукупності, тобто $x_1 = -1$ (точка K), і корінь третього рівняння сукупності, тобто $x_2 = 3$ (точка L).

Якщо $a < -3$ (пряма A_5B_5), то заданому рівнянню задовільняють два корені: один корінь з першого рівняння сукупності, тобто $x_1 = a/3$ (точка M), і корінь з третього рівняння сукупності, тобто $x_2 = -a$ (точка N).

Розроблення програмно-демонстраційного комплексу.

Основою програмного комплексу є реалізована у середовищі C++Builder програмний продукт, що складається з семи форм, одна з яких є головною і виконує роль головного меню (рис. 4). Як бачимо з вигляду головної форми, на ній розміщено ряд командних кнопок, призначення кожної з яких – реалізувати ту чи іншу задачу проекту, тобто – розв'язати те чи інше рівняння. Сам же розв'язок забезпечується розробленим нами програмним кодом. На рис. 4 та рис. 5 показано форми, які забезпечують виклад теоретичного матеріалу та реалізацію методів.

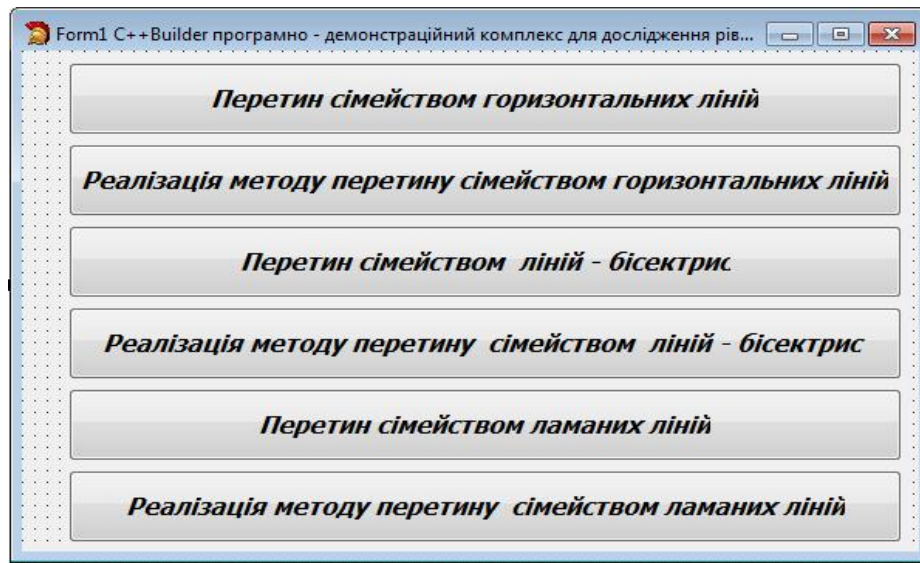


Рис. 3 – Головна форма проекту

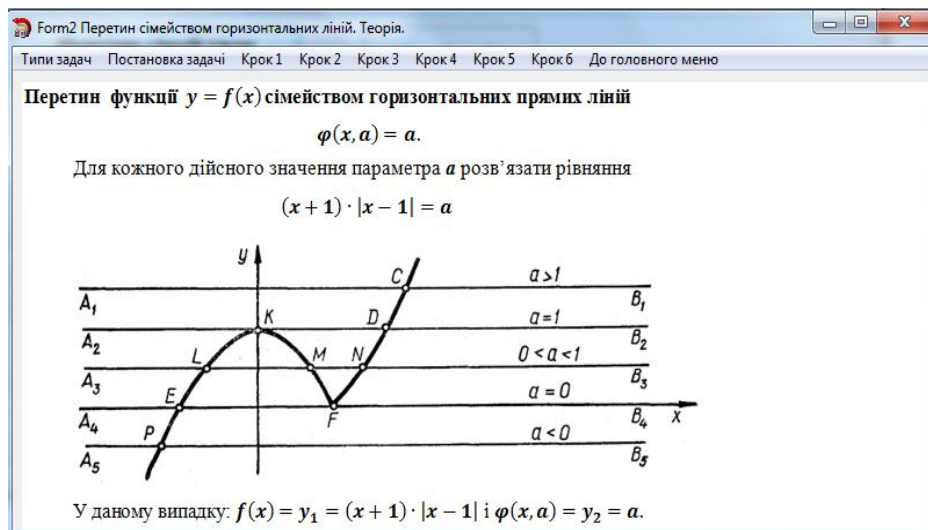


Рис. 4 – Форма проекту з теоретичними викладками

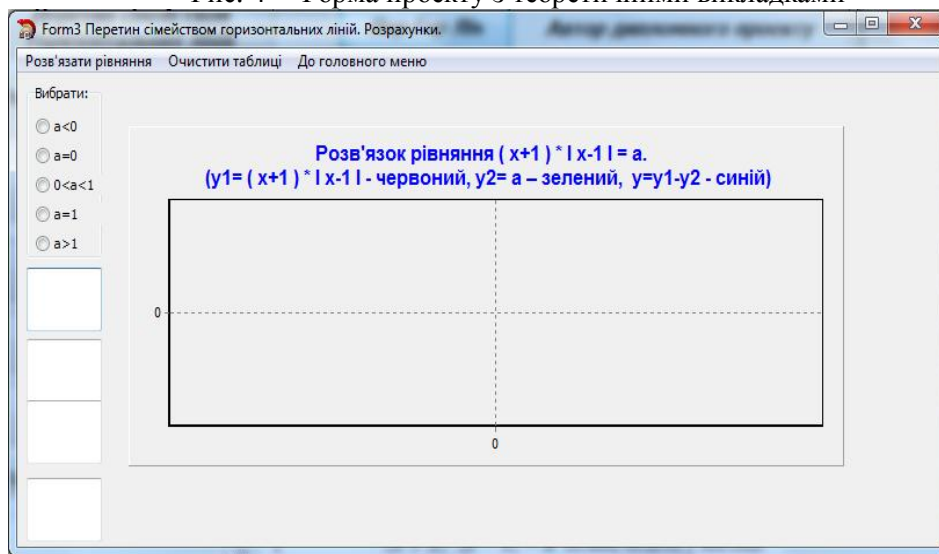


Рис. 5 – Форма проекту з реалізацією методу

Результати розв'язування рівняння $(x+1) \cdot |x-1| = a$ методом перетинів горизонтальними лініями (рис. 6 - рис. 10).

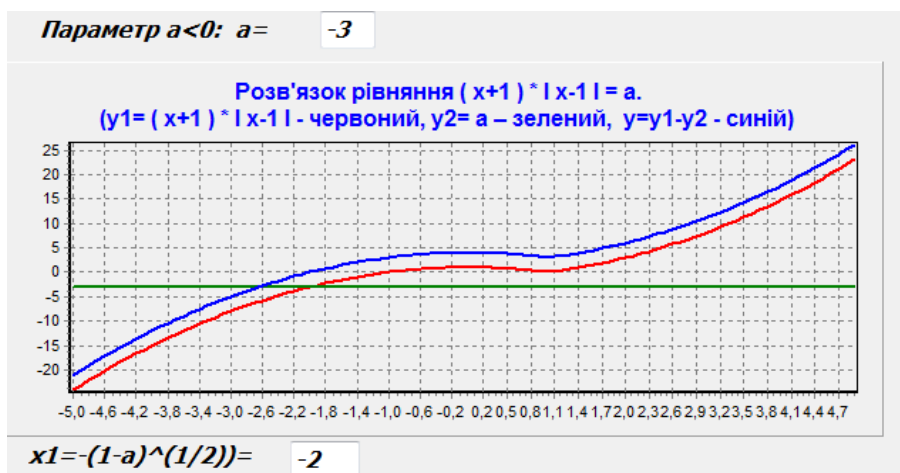


Рис. 6 – Розв'язок рівняння $(x+1) \cdot |x-1| = a$, якщо $a < 0$.

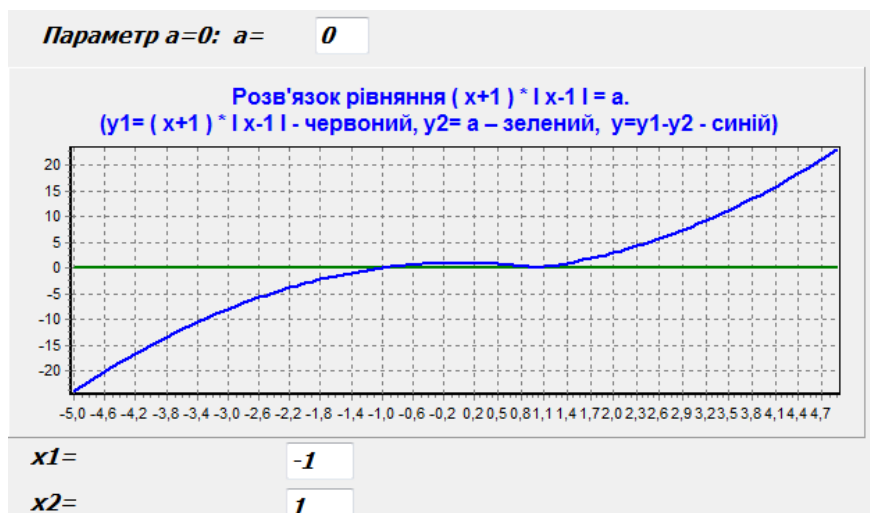


Рис. 7 – Розв'язок рівняння $(x+1) \cdot |x-1| = a$, якщо $a = 0$.

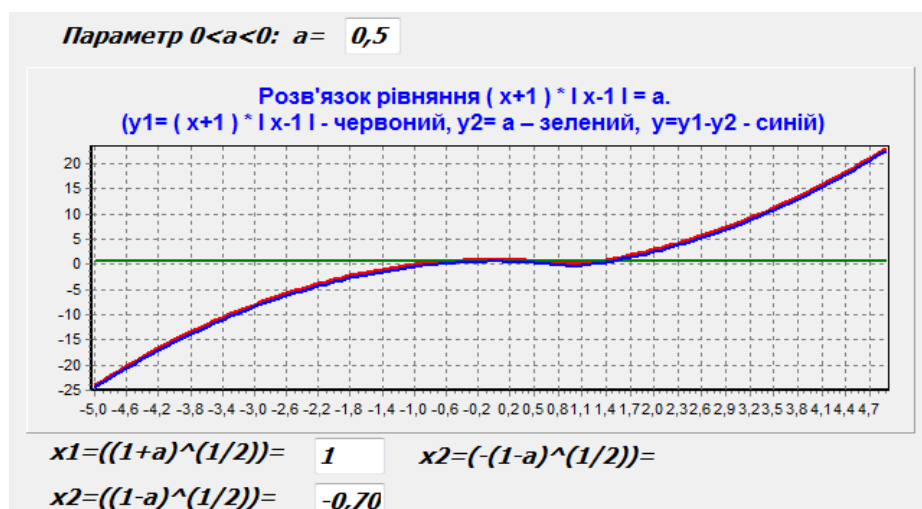


Рис. 8 – Розв'язок рівняння $(x+1) \cdot |x-1| = a$, якщо $0 < a < 1$.

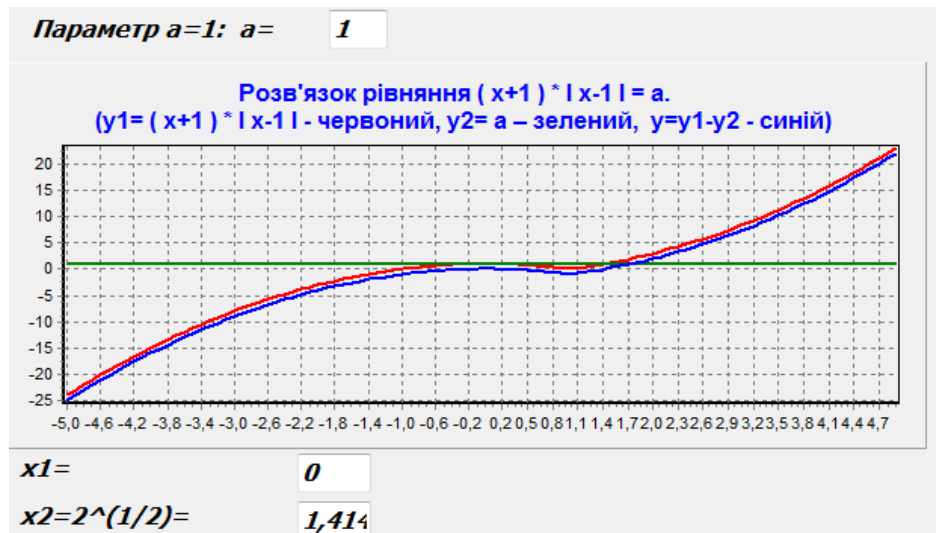


Рис. 9 – Розв'язок рівняння $(x+1) \cdot |x-1| = a$, якщо $a = 1$.

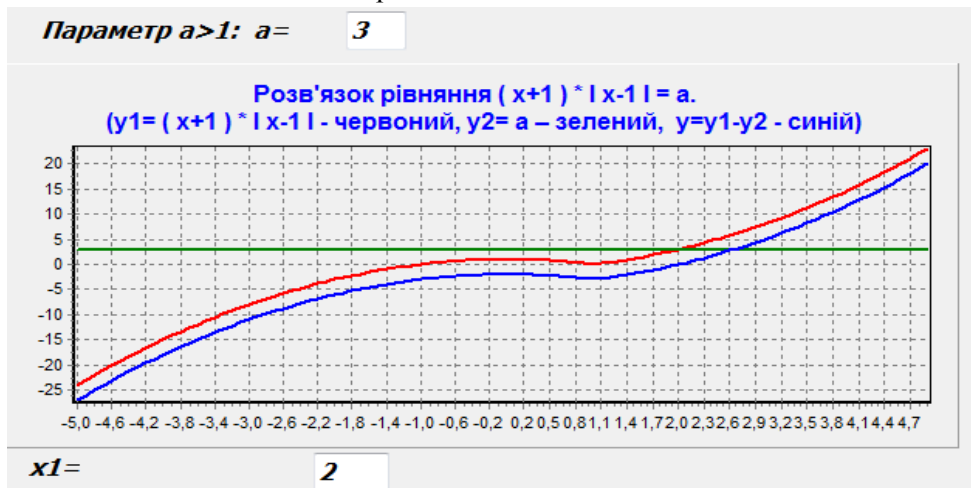


Рис.10 – Розв'язок рівняння $(x+1) \cdot |x-1| = a$, якщо $a > 1$.

Результати розв'язування рівняння $|x-1| + |x+1| = x-a$ методом перетинів бісекторними лініями (рис. 11 - рис. 15).

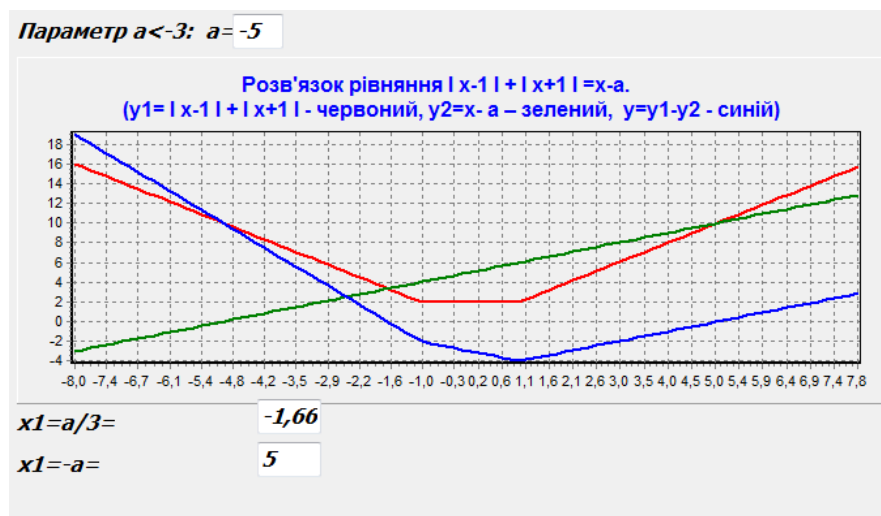


Рис. 11 – Розв'язок рівняння $|x-1| + |x+1| = x-a$, якщо $a < -3$.

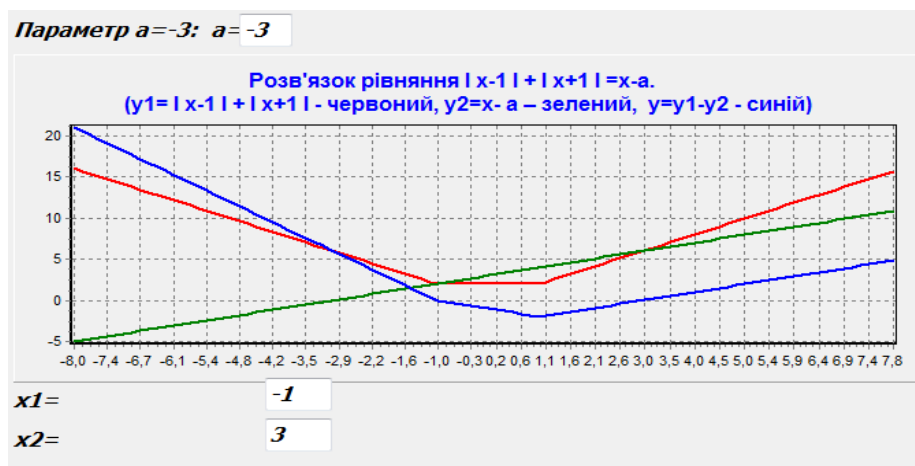


Рис. 12 – Розв'язок рівняння $|x-1| + |x+1| = x-a$, якщо $a = -3$.

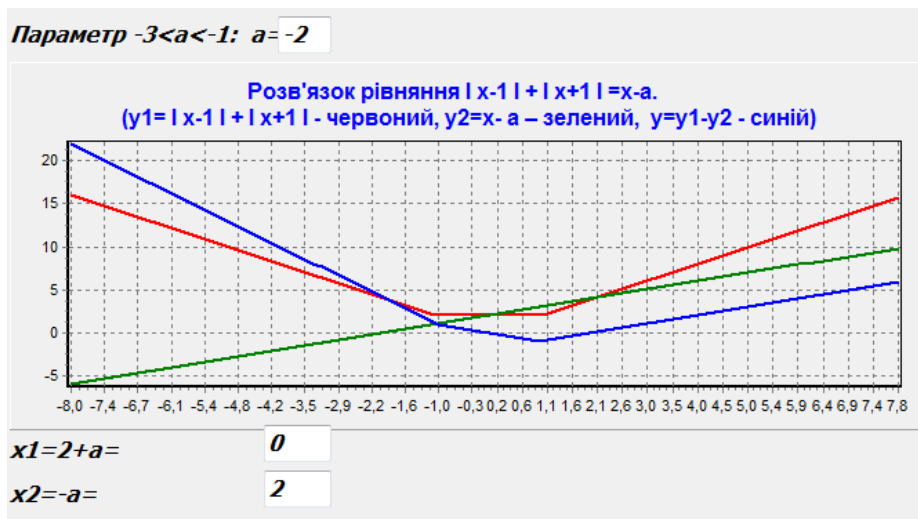


Рис.13 – Розв'язок рівняння $|x-1| + |x+1| = x-a$, якщо $-3 < a < -1$.

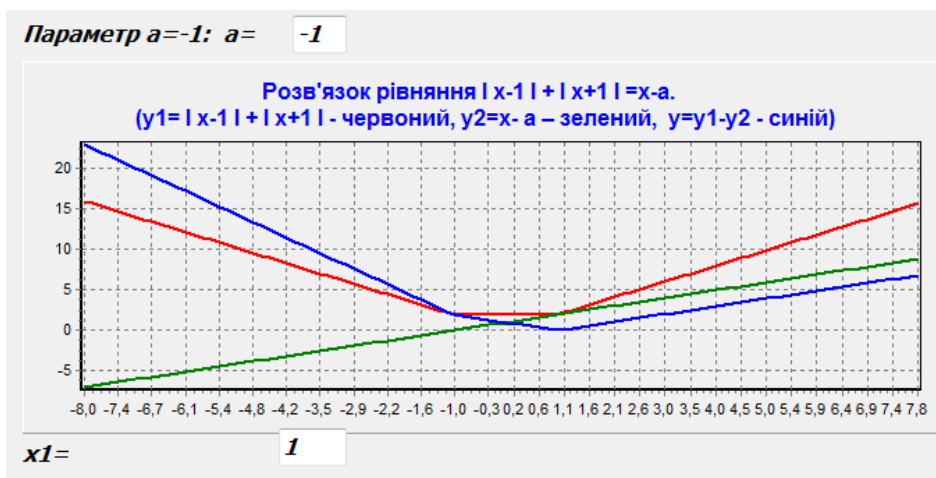


Рис.14 – Розв'язок рівняння $|x-1| + |x+1| = x-a$, якщо $a = -1$.

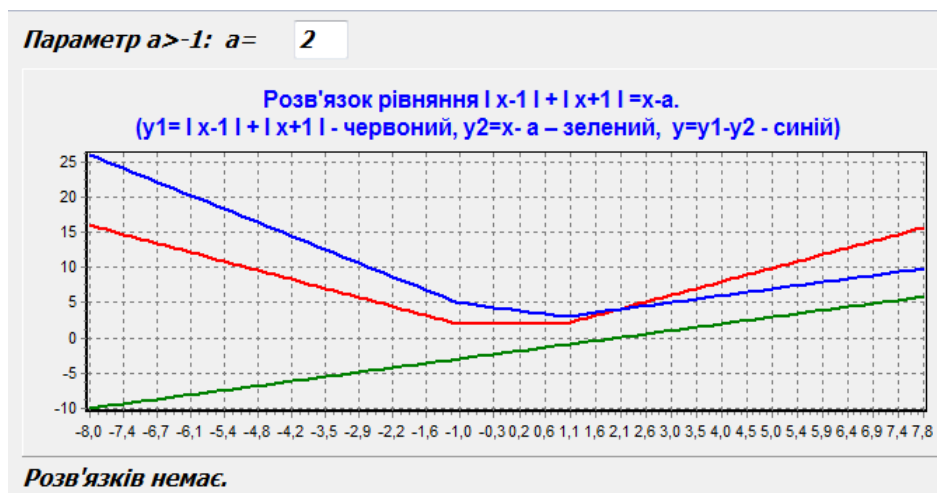


Рис.15 – Розв'язок рівняння $|x-1| + |x+1| = x-a$, якщо $a > -1$.

Висновки.

1. В роботі розроблено C++Builder програмно-демонстраційний комплекс, у якому сучасними програмними засобами розв'язується актуальна задача дослідження рівнянь з параметрами методом перетинів прямими лініями з залученням розробленого авторами програмно-демонстраційного комплексу. Ця задача полягає у визначенні коренів рівняння за різних можливих значень параметрів. Наведені окремі результати такого дослідження.

2. Розроблений C++Builder програмно-демонстраційний комплекс може бути використаний як в навчальному процесі при викладанні відповідної теми, так і з метою більш швидкого і ефективного розв'язування рівнянь з параметрами, які традиційно викликають певні труднощі.

1. Шарова Л.И. Уравнения и неравенства: Пособие для подготовительных отделений. –К.:Вища школа, 1981.-290 с.
2. Прус А.В., Швець А.О. Задачі з параметрами в шкільному курсі математики. Навчальний посібник. –Житомир: «Рута», 2016. -468 с.
3. Мирошин В.В. Решение задач с параметрами. Теория и практика. М.: Экзамен, 2009. – 286 с.
4. Горнштейн С.В., Полонський В.Б., Якір М.С. Завдання з параметрами. К.: РІА Текст: МП ОКО, 1992. – 290 с.
5. Пех П.А. Програмний C++ комплекс та результати дослідження швидкодії алгоритмів сортування методом імітаційного моделювання / Бортник К.Я., Яручик О.В.// Комп'ютерно- інтегровані технології: освіта, наука, виробництво: Науковий журнал. –Луцьк: РВВ ЛНТУ, 2017.Випуск № 27. ст.. 54 – 60.
6. Пех П.А. Програмний C++ комплекс для дослідження процесу функціонування автоматизованих ліній та його верифікація / Войтович А.О.// Комп'ютерно-інтегровані технології: освіта, наука, виробництво: Науковий журнал. –Луцьк: РВВ ЛНТУ, 2017.Випуск № 27. ст.. 60 – 66.

УДК 004.415.3

Пех П.А., Тихомиров В.В

Луцький національний технічний університет

МЕТОДИ ФОРМУВАННЯ СИСТЕМНИХ ВИМОГ ДО АЗ ДЛЯ ГРОМІЗДКИХ 3D ПРОГРАМ

Пех П.А., Тихомиров В.В. Методи формування системних вимог до АЗ для громіздких 3D програм. У статті розкрито важливість встановлення мінімальних та номінальних системних вимог для коректного відображення додатку і забезпечення найкращих умов роботи користувача з даною програмою. Запропоновано спосіб отримання даних щодо швидкодії певних систем при роботі з конкретним, раніше підготовленим додатком.

Ключові слова: швидкодія, системні вимоги, додаток, дані.

Пех П.А., Тихомиров В.В. Методы формирования системных требований к АО для громоздких 3D программ. В статье раскрыта важность установления минимальных и рекомендуемых системных требований для корректного отображения приложения и обеспечения наилучшего опыта пользователя работы с данной программой. Предложен способ получения данных по быстродействию определенных систем при работе с конкретным, ранее подготовленным приложением.

Ключевые слова: быстродействие, системные требования, приложение, данные.

Pekh P.A., Tykhomyrov V.V. Methods for forming system requirements to AS for large 3D programs. In the article is shown the importance of determining minimum and nominal system requirements for the correct application display and ensuring the best conditions for the user to work with it. The method of obtaining data on the performance of certain systems is offered when working with a specific, previously prepared application.

Keywords: performance, system requirements, application, data.

Постановка проблеми. В наш час розробляється багато власних комп'ютерних програм (додатків), для роботи з якими використовуються різні конфігурації апаратного забезпечення (АЗ). Кожен такий програмний продукт потребує певного обсягу ресурсів робочої станції — чи то персонального комп'ютера, чи іншого пристрою. Залежно від типу та складності додатку змінюється навантаження конкретних ресурсів апаратної складової комп'ютера — процесора, оперативної пам'яті, відеокarti, жорсткого диску та інших. Визначитися з тим, чи здатна робоча станція ефективно працювати з конкретним програмним додатком, не викликаючи при цьому дискомфорту користувача, мають допомогти системні вимоги. Ці системні вимоги формує розробник для свого програмного забезпечення (ПЗ).

У даній статті розглядається проблема визначення мінімальних та номінальних системних вимог на прикладі розробленого нами програмного продукту, що пов'язане з аналізом даних отриманих емпірично, та за допомогою математичних розрахунків.

Виходячи з поставлених задач, виникають питання, які стосуються проблеми збору даних з програмного продукту та представлення їх у вигляді таблиці. Окрім вищезазначеної проблеми існує також питання вибору програмного забезпечення для обробки зібраних даних та винесення вердикту, отримання кінцевого результату у вигляді мінімальних та рекомендованих системних вимог.

Перш ніж розглянути питання збору даних з програмного забезпечення необхідно визначитись з:

- піддослідними стендами, тобто ПК, на яких будуть проводитись дослідження;
- ПЗ, для якого формуватимуться системні вимоги;
- алгоритмом визначення системних вимог;
- ПЗ для вимірювання необхідних даних, пов'язаних з навантаженням певних вузлів апаратної частини;
- ПЗ для аналізу отриманих даних та представлення його у прийнятному для користувача вигляді.

Аналіз останніх досліджень та публікацій. Методи тестування програмного забезпечення є надзвичайно актуальною і розглядалися багатьма дослідниками [1 - 5], однак жоден з них не розглядав проблему формування системних вимог до АЗ для конкретних додатків.

Для того, щоб розпочати процес отримання необхідних даних, слід визначитись з апаратним забезпеченням, яке буде використовуватись для дослідження. Необхідно мати хоча б 3 — 4 робочих станції різної конфігурації. Тоді віднаходження “золотої середини” в параметрах

швидкодії додатку буде значно простішим. Вибір апаратного забезпечення здійснюється на основі аналізу багатьох факторів, проте вирішальною в нашому випадку стала сама наявність конкретних робочих станцій. Для даного дослідження було вирішено обрати чотири ПК, що були у нас у наявності, з наступними конфігураціями (таблиця 1).

Таблиця 1 — Конфігурації робочих станцій для тестування програмного забезпечення

Елемент АЗ	Назва стенду			
	N1(1050)	N2(1050ti)	N3(630M)	N4(630HD)
ЦПУ	Pentium g4600	Pentium g4600	Pentium B980	I7 7700
ГПУ	gtx 1050	gtx 1050ti	gt 630M	Intel HD 630
ОЗП	8 GB 2133 MHz	16 GB 2133 MHz	8 GB 1333 MHz	16 GB 2133 MHz

Визначившись з робочими станціями для тестування, необхідно обрати ПЗ, на основі якого будуть проводитись дослідження і формуватися системні вимоги. Як відомо, в даний час існує достатня кількість “тяжких” та “легких” програмних продуктів, тому вибір великий. Одним з головних критеріїв вибору об’єкта дослідження є необхідність його тестування з метою формування системних вимог. Нами для дослідження обрано додаток власного виробництва — “3D модель замку Любарта”. [3]

Далі слід перейти до наступного етапу планування дослідження, а саме вибору методики проведення цього дослідження. Існує два основних шляхи досягнення результату розробника ПЗ, який прагне визначити системні вимоги для свого додатку, це:

- емпіричний;
- статистичний.

Під чисто емпіричним дослідженням розуміється проведення низки практичних тестів, які і вкажуть на вирішення проблеми, допоможуть визначити необхідні характеристики для комп’ютера користувача під певний додаток. Проте цей шлях часто є мало ефективним, адже вимагає практичного виконання програми на всіх робочих станціях, а це може зайняти велику кількість часу, що, в деяких випадках, є неприпустимим для розробника, який працює з ПЗ на останній стадії розробки — тестуванні. Проте безумовним плюсом цього методу є перевірка додатку вживу, без використання математичних розрахунків та штучного підбору ідеального АЗ. Розробник має можливість прямо контактувати зі своїм додатком в процесі роботи та аналізувати ефективність використання того чи іншого тиу АЗ.

Існує також інший тип дослідження — статистичний. Звичайно, невід’ємною частиною статистичного аналізу є емпіричне дослідження, але воно проводиться за певним планом з метою отримання необхідних статистичних даних з метою їх подальшого оброблення. Мінусом цього методу є те, що в результаті оброблення ми отримуємо середньостатистичні дані, однак у разі використання не лише середніх значень, а й дисперсії досліджуваних параметрів, ми маємо змогу об’єктивно підходити до процесу формування системних вимог. Плюсом даного методу є значне прискорення процесу тестування та отримання результатів, підтверджених об’єктивними розрахунками. В даній статті буде розглянуто обидва методи аналізу даних.

Після того, як розробник обрав метод дослідження швидкодії програмного продукту, необхідно звернути увагу на ПЗ, завдяки якому можна буде отримувати дані про стан апаратних частин робочої станції в певний період часу. Серед таких програм є: MSI Afterburner, та платна Aida 64. Здійснювати Моніторинг АЗ з допомогою цих програм здійснюється за певні проміжки часу. Критерієм вибору ПЗ для вищевказаних цілей є: легкість у використанні, мала вартість, рівень адаптованості до поставленої задачі та особистий досвід користування. Зваживши всі варіанти, враховуючи безкоштовність та зручність у використанні, було обрано програму MSI Afterburner. Вона дозволяє здійснювати моніторинг системи в режимі реального часу, не згортаючи при цьому додаток.

Маючи в розпорядженні робочі станції, обраний метод дослідження та програму для моніторингу, слід здійснити аналіз щодо ПЗ для роботи з отриманими числовими даними та їх інтерпретування у зрозумілу форму навіть для звичайного користувача. Найбільш популярними

серед таких програм є: Microsoft Office Excel, LibreOffice Calc, Google Tablets, доступна з будь якого браузера. Критеріями вибору одного з вище перерахованих додатків є: вартість, простота у використанні, набір необхідних інструментів, особистий досвід. Всі програми з даного списку володіють необхідним функціоналом для задоволення наших потреб та зручні у використанні. За доступністю безумовним лідером є Google Таблиці, проте вибір було зроблено на користь програми Microsoft Office Excel, через великий досвід роботи з нею та наявність її на комп'ютері, де проходитиме збір усіх даних та подальший їх аналіз.

Основна частина. На цьому кроці етап планування закінчений і необхідно переходити до наступного — збору даних, потрібних для визначення системних вимог додатку “3D модель замку Любарта”. Першим кроком буде встановлення програми MSI Afterburner на всі робочі станції: N1 1050, N2 1050ti, N3 630M, N4 630 HD. Далі необхідно провести налаштування цього додатку для коректного відображення інформації під час тестування (рис. 1а, рис. 1б, рис. 1в).

Після проведення всіх налаштувань, можна запускати програму “3D модель замку Любарта”. Одразу після запуску додатка в лівому верхньому куті з'явиться вся необхідна для тестування інформація (рис. 2).

Впевнившись в тому, що зчитування даних працює належним чином, можна переходити з головного меню в гру. Для прикладу, були взяті значення з комп'ютера №3 (630M) (рис. 3).

Ознайомившись з територією замку на протязі хвилини, були записані значення CPU load, GPU load, FPS. Інтервал між замірами — 3 секунди. В результаті був отриманий наступний графік значень (рис. 4).

Середні значення вищевказаних показників: CPU load — 55.8%, GPU load — 98.1%, FPS — 14.8. Дисперсія показників: CPU load — 7.32, GPU load — 1.04, FPS — 0.27. Дисперсія протестованих показників представлена нижче (рисунок 5).

Судячи з отриманих даних, можна зазначити, що для комфортного відтворення додатку представлена вище конфігурація в цілому не підходить, оскільки середнє значення FPS (frames per second) в районі 14.8 не є прийнятним для користувача. Слід зауважити, що прийнятний рівень показника кадрів за секунду становить від 25 до 60.

Вузьким місцем представленої вище системи N3 630M є відеочіп, в той час як процесор завантажений в середньому на 55,8%. Проте високий рівень дисперсії значень завантаження процесора у відсотках говорить про те, що він працює нестабільно. Також були проведені тести на інших системах, результати яких описані нижче (рис. 6а, рис. 6б, рис. 6в).

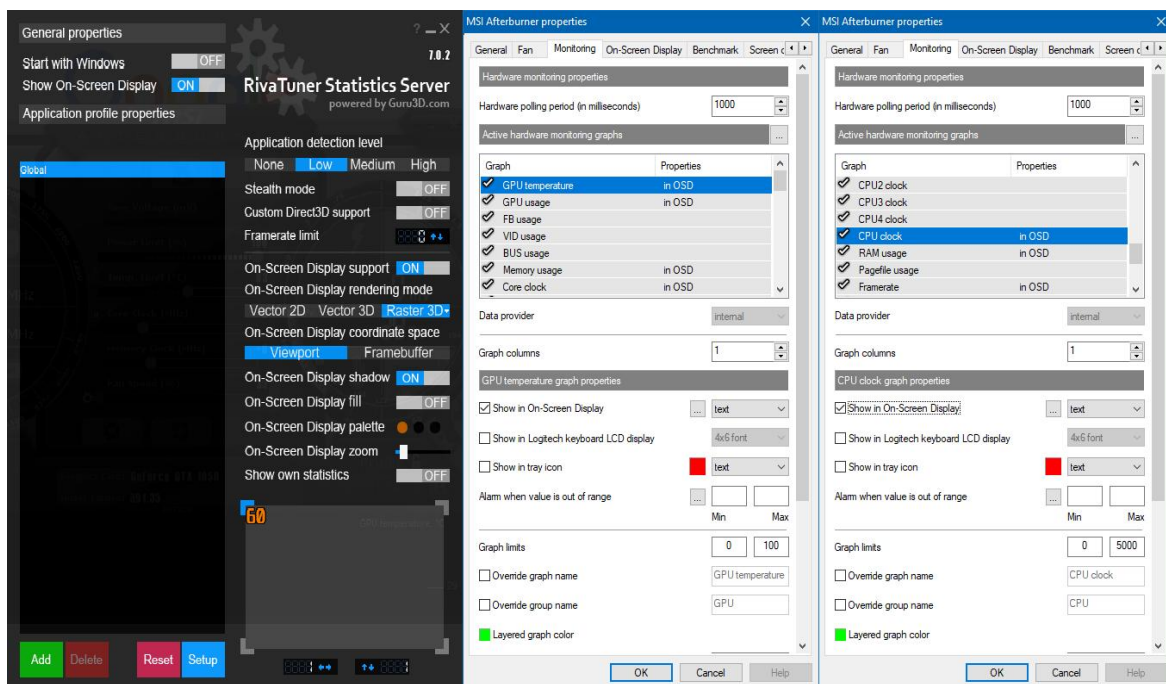


Рисунок 1а — Налаштування MSI Afterburner

Рисунок 1б — Налаштування MSI Afterburner Monitoring

Рисунок 1в — Налаштування MSI Afterburner Monitoring

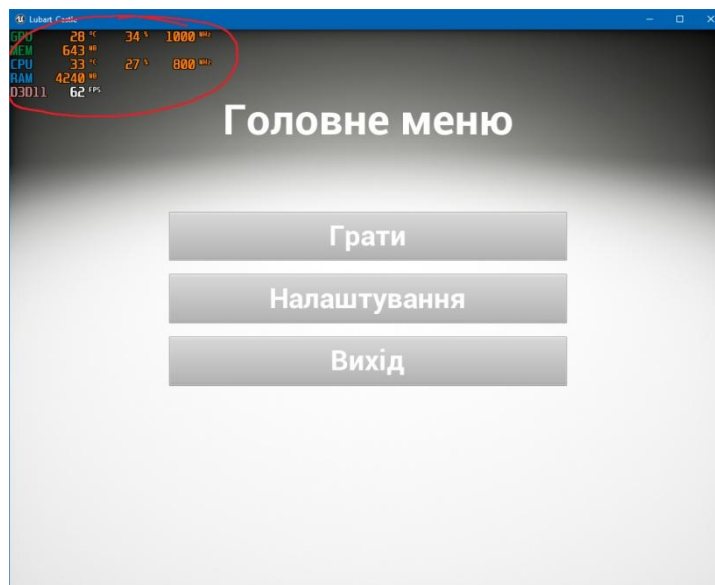


Рисунок 2 — MSI Afterburner під час виконання програми, яку тестують



Рисунок 3 — Зображення з гри

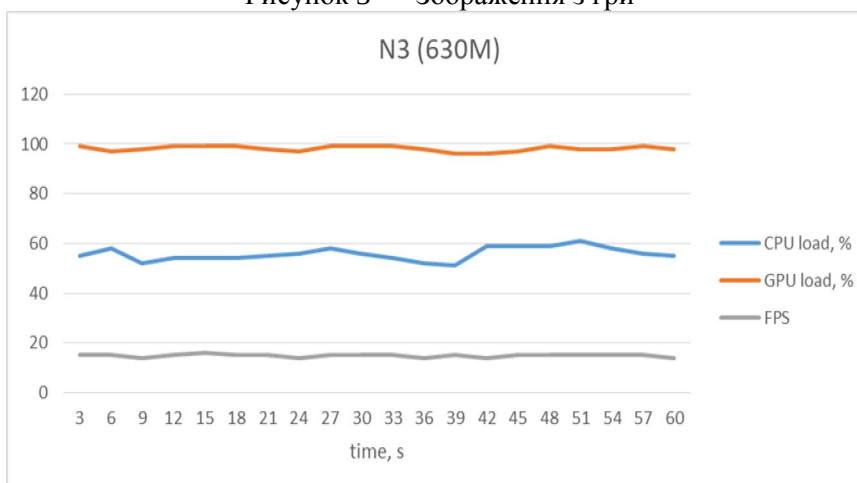


Рисунок 4 — Графік значень протестованих параметрів стенда N3 (630M)

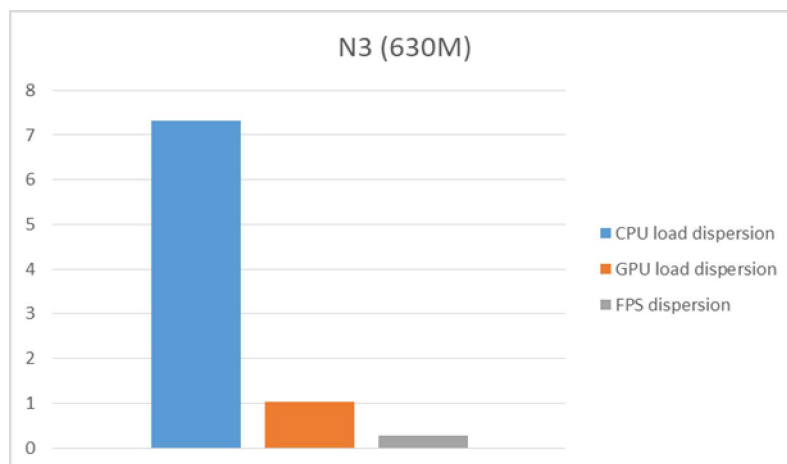


Рисунок 5 — дисперсія протестованих значень з N3 (630M)

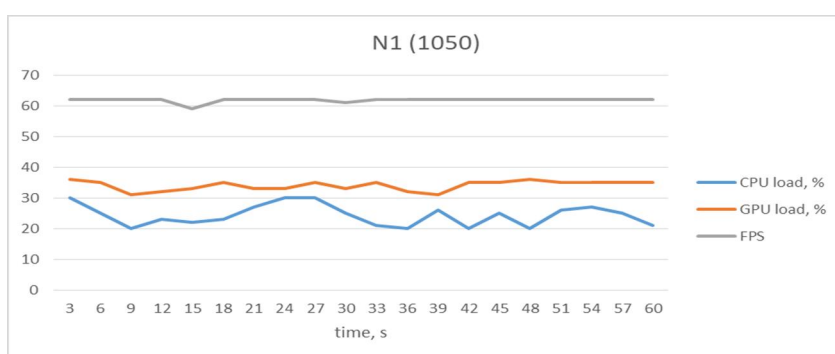


Рисунок 6а — результат тестування системи N1 (1050)

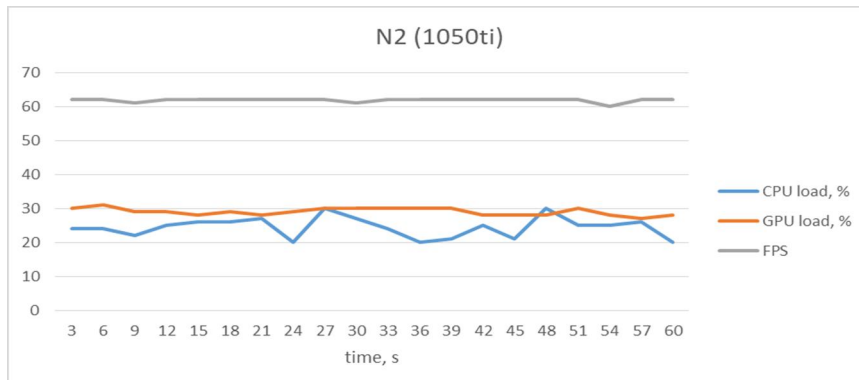


Рисунок 6б — результат тестування системи N2 (1050ti)

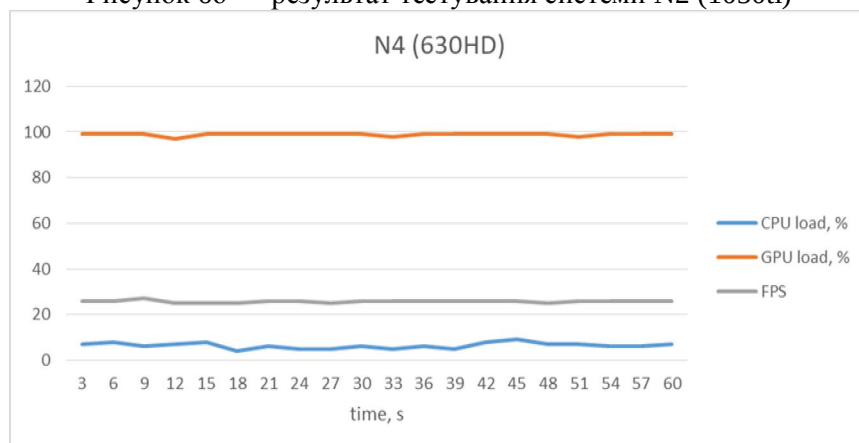


Рисунок 6в — результат тестування системи N4 (630HD)

Спираючись на отримані тести, можна зазначити, що для мінімальних системних вимог додатку допустимим буде рівень процесору зі стенда N3 630M, адже він показав невисокий рівень завантаженості. В той же час відеокарта 630M для мінімальних системних вимог занадто слабка, тому слід звернути увагу на GPU рівня Intel 630HD, що забезпечить системі прийнятний рівень швидкодії у вигляді 24 – 26 кадрів на секунду.

Вище був представлений приклад аналізу даних для виявлення мінімальних системних вимог додатку. Далі буде коротко описано теоретичний метод отримання системних вимог, але вже не мінімальних, а номінальних.

116	 DX11	ATI Radeon HD 5850 ★★★★☆	\$324	2260
117	 DX11	AMD Radeon HD 7750 ★★★★☆	\$110	2240
118	 DX11	NVIDIA GeForce GT 755M ★★★★☆	N/A	2160
119	 DX11	AMD Radeon HD 6790 ★★★★☆	\$145	2150

Рисунок 7 — Перелік відеокарт, коефіцієнт швидкодії (згідно з futuremark) яких приблизно відповідає заданим вимогам

Номінальні системні вимоги додатка вказують на характеристику A3, рівня потужності якого достатньо для досягнення найкомфортнішого відтворення цього додатка. Щоб одержати синтетичну оцінку відеокарти, слід звернутись до бази даних процесорів та відеочіпів на сайті <https://www.futuremark.com/hardware/gpu>. Вибраши 630M, побачимо коефіцієнт швидкодії 530. Нехай це число 530 відповідає середній частоті кадрів 15 для нашого додатка. Для отримання значення орієнтовно 60, що означатиме комфортний режим роботи додатка, слід помножити 15 на 4, і, відповідно, коефіцієнт 530 також варто помножити на 4. Отримаємо відеочіп, коефіцієнт швидкодії якого приблизно відповідає значенню 2120 (рисунок 7).

Висновки та рекомендації. Таким чином, для отримання системних вимог до A3 під конкретний додаток необхідно вибрати різні типи робочих станцій, встановити на них необхідне ПЗ, протестувати роботу додатку через певні проміжки часу. Запропоновані два основних шляхи формування системних вимог до A3 під конкретний додаток – емпіричний та статистичний.

1. Лайза Криспин, Джанет Грегорі. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд = Agile Testing: A Practical Guide for Testers and Agile Teams.— М.: «Вильямс», 2010. — 464 с.
2. Канер Кем, Фолк Джек, Нгуен Енг Кек. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. — Киев : ДиаСофт, 2001. — 544 с.
3. Калбертсон Роберт, Браун Крис, Кобб Гэри. Быстрое тестирование. — М. : «Вильямс», 2002. — 374 с.
4. Сеницын С. В., Налютин Н. Ю. Верификация программного обеспечения. — М. : БИНОМ, 2008. — 368 с.
5. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. — СПб. : Питер, 2004. — 320 с.

УДК 004.051

Попов Д. А., Костенко В. Л., Ядрова М. В., Кондратьев С. Б.
Одесский национальный политехнический университет

РАЗРАБОТКА МУЛЬТИСЕНСОРНОГО СОНАРА ДЛЯ СЛЕПЫХ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА STM8L

Попов Д. А., Костенко В. Л., Ядрова М. В., Кондратьев С. Б. Разработка мультисенсорного сонара для слепых на основе микроконтроллера STM8L. В данной статье приведены результаты разработки мультисенсорного сонара для слепых. Показана возможность расширения угла обзора, снижения энергопотребления за счет применения современной элементной базы, в том числе микроконтроллеров с минимальным энергопотреблением. Разработан алгоритм работы микроконтроллера и код программы на языке C++. Представлен макет устройства.

Ключевые слова: сонар для слепых; микроконтроллер; звуковой излучатель; программные коды; энергопотребление.

Попов Д. О., Костенко В. Л., Ядрова М. В., Кондратьев С. Б. Розробка мультисенсорного сонара для сліпих на основі мікроконтролера STM8L. В даній статті наведені результати розробки мультисенсорного сонара для сліпих. Показана можливість розширення кута огляду, зниження енергоспоживання за рахунок застосування сучасної елементної бази, в тому числі мікроконтролерів з мінімальним енергоспоживанням. Розроблено алгоритм роботи мікроконтролера і код програми на мові C++. Представлений макет пристрою.

Ключові слова: сонар для сліпих; мікроконтроллер; звуковий випромінювач; програмні коди; енергоспоживання.

Popov D., Kostenko V., Yadrova M., Kondratyev S. Development of a multisensory sonar for the blind based on the STM8L microcontroller. This article presents the results of developing a multisensory sonar for the blind. The possibility of expanding the viewing angle, reducing power consumption through the use of a modern element base, including microcontrollers with minimal power consumption, is shown. The algorithm of the microcontroller and the code of the program in C++ are developed. The device layout is presented.

Keywords: sonar for the blind; microcontroller; voice emitter; codes of programs; energy consumption.

Для решения важной социальной проблемы – реабилитации слабовидящих и незрячих людей требуются разработки новых и совершенствования известных средств обнаружения травматически потенциальных препятствий, находящихся в пространстве перед ними. Среди таких средств в настоящее время большое внимание отводится мобильным ультразвуковым устройствам с микроконтроллерами, интегрированными на одной плате с миниатюрными датчиками и устройствами индикации [1, 2, 3].

В настоящее время наибольшую популярность получили ультразвуковые сонары на базе микроконтроллера Arduino [4, 5]. При эффективном угле наблюдения до 15° и дальности до 1,5 м, устройство потребляет ток до 140 мА, при этом расстояние до препятствия не определяется, а среднее время автономной работы на одном заряде элемента питания составляет 8 часов. Для повышения локальности обнаружения, в работе [6] предлагаются ультразвуковые перчатки для слепых, пригодные для обнаружения объектов на расстоянии до 3,5 м с эффективным сектором наблюдения не менее 30 градусов и с интегрированным процессором Arduino Pro Mini 328. При энергопотреблении от 33 мВт до 100 мВт, время автономной работы устройства составляет 4–8 часов в зависимости от используемой платы. Устройство удобно для использования внутри помещений, однако не оптимально на улице из-за необходимости постоянно направлять руки с перчатками в сторону предполагаемых препятствий. Для обнаружения препятствий на уровне подстилающей поверхности и расстояния до них в [7] предлагается ультразвуковая «умная трость» для незрячих на базе микроконтроллера семейства Arduino. Обладая расширенными функциональными возможностями данное устройство, вместе с тем, обеспечивает невысокий эффективный угол обзора – порядка 14–30 градусов и обладает недостаточно большим сроком работы в автономном режиме без подзарядки элемента питания. С целью повышения информативности обнаружения, в работе [8] предлагается использование системы обнаружения и позиционирования с применением трех ультразвуковых сенсоров для определения препятствия на пути слепых в трех различных направлениях, GPS и система сообщения СМС, микроконтроллер Arduino, зуммер и генератор вибраций. Недостатком систем обнаружения препятствий для слепых на базе этих плат является их сравнительно высокое энергопотребление и, соответственно, малое время работы на одном заряде элемента питания, необходимость частого заряда устройства.

Проведенные нами исследования показали, что одним из основных факторов, ограничивающих применение известных сонаров для слепых, является большое потребление мощности микроконтроллером Arduino, которое, в зависимости от применяемой платы, меняется

от 33 мВт (плата Arduino LilyPad USB, питание от аккумулятора 3,7 В) до 105 мВт (плата Beetle leonardo, при питании от аккумулятора 5 В) [9].

Поэтому исследования по энергосбережению сонаров для слепых являются важной научно-прикладной **проблемой**, которая может быть решена на основе применения современной элементной базы, в том числе микроконтроллеров с минимальным энергопотреблением [11, 12], и разработки соответствующего программного обеспечения.

Изложение основного материала. С учетом проведенных исследований были разработаны программное обеспечение и макет сонара для слепых, который включает два ультразвуковых сенсора HC-SR04, микроконтроллер STM8L с низким энергопотреблением, разработанный специально для изделий с батарейным питанием, и стандартный малогабаритный элемент питания на 3 В. Конструктивно предусмотрена возможность заряда элемента питания через USB интерфейс. В качестве индикатора направления на препятствие и расстояния до него, с учетом минимизации энергозатрат, был выбран звуковой излучатель Piezo Buzzer 17855 с током потребления не более 2 мА.

Программным путем обеспечивается последовательный опрос сенсоров и возможность перехода неактивного датчика в режим тишины до следующего акта измерений с целью снижения энергопотребления и экономии заряда батареи.

Схема алгоритма работы микроконтроллера с двумя сенсорами представлена на рис. 1.



Рисунок 1 – Алгоритм работы микроконтроллера с двумя сенсорами

В процессе работы сонара микроконтроллер формирует короткий запускающий импульс, который подается поочередно на вход каждого ультразвукового датчика. На выходе датчиков получаем импульс, длительность которого прямо пропорциональна расстоянию до объекта. Микроконтроллер преобразует полученный импульс в серию импульсов, длительность которых постоянна, а частота следования обратно пропорциональна расстоянию до объекта. Эти импульсы поступают на один из звуковых излучателей, смонтированных на плате совместно с соответствующим датчиком. По частоте следования звукового сигнала можно ориентировочно определить удаленность до объекта на слух, а по номеру звукового излучателя – направление на препятствие.

Код программы написан на языке C++. Основной фрагмент кода программы представлен на рис. 2. Звуковые излучатели расположены в блоке с датчиками и индицируют расстояние до препятствия паузами между звуковыми сигналами. Индикация заканчивается в тот момент, когда индикация самого длинного расстояния будет произведена, после этого программа производит переход на начало измерений.

```

void UltrasonicSensor::BeepersActivation()
{
    if(echoLengthFirstSensor <= echoBackFront && echoLengthSecondSensor <= echoBackFront)
    {
        distanceFirstObstacle = echoLengthFirstSensor / soundSpeed;
        distanceSecondObstacle = echoLengthSecondSensor / soundSpeed;
        if(distanceFirstObstacle <= distanceOneMeter && distanceSecondObstacle <=
            distanceOneMeter)
        {
            if(distanceFirstObstacle < distanceSecondObstacle)
            {
                /* Determine the number of beeps */
                uint32_t temp = (echoLengthSecondSensor * delayAfterBeep + delayBetweenBeep)
                / (echoLengthFirstSensor * delayAfterBeep + delayBetweenBeep);
                /* Activating the beeper */
                GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_SET);
                Delay(delayBetweenBeep);
                GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_RESET);
                for(uint32_t i = 0; i < temp+1; ++i)
                {
                    /* Activating the beeper */
                    GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_SET);
                    Delay(delayBetweenBeep);
                    GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_RESET);
                    Delay(echoLengthFirstSensor * delayAfterBeep);
                }
            }
            else if(distanceSecondObstacle <= distanceFirstObstacle)
            {
                /* Determine the number of beeps */
                uint32_t temp = (echoLengthFirstSensor * delayAfterBeep + delayBetweenBeep)
                /
                (echoLengthSecondSensor * delayAfterBeep + delayBetweenBeep);
                /* Activating the beeper */
                GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_SET);
                Delay(delayBetweenBeep);
                GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_RESET);
                for(uint32_t i = 0; i < temp+1; ++i)
                {
                    /* Activating the beeper */
                    GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_SET);
                    Delay(delayBetweenBeep);
                    GPIO_WriteBit(GPIOC, GPIO_Pin_11,
                    Bit_RESET);
                    Delay(echoLengthSecondSensor * delayAfterBeep);
                }
            }
        }
    }
}

else if(echoLengthFirstSensor <= echoBackFront && echoLengthSecondSensor > echoBackFront)
{
    distanceFirstObstacle = echoLengthFirstSensor / soundSpeed;
    if(distanceFirstObstacle <= distanceOneMeter)
    {
        /* Activating the beeper */
        GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_SET);
        Delay(delayBetweenBeep);
        GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_RESET);
        Delay(echoLengthFirstSensor * delayAfterBeep);
    }
}

else if(echoLengthFirstSensor > echoBackFront && echoLengthSecondSensor <= echoBackFront)

```

```

{
    distanceSecondObstacle = echoLengthSecondSensor / soundSpeed;
    if(distanceSecondObstacle <= distanceOneMeter)
    {
        /* Activating the beeper */
        GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_SET);
        Delay(delayBetweenBeep);
        GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_RESET);
        Delay(echoLengthSecondSensor * delayAfterBeep);
    }
}
    
```

Рисунок 2 – Основной фрагмент кода программы

Внешний вид макета сенсора с двумя датчиками представлен на рис. 3



Рисунок 3 – Макет сенсора с двумя ультразвуковыми датчиками

Ультразвуковые датчики расположены под углом друг к другу. Такое расположение датчиков по сравнению с аналогами [4], обеспечивает увеличение эффективного угла обзора в горизонтальной плоскости.

Предложенное программное обеспечение позволяет наращивать количество датчиков и звуковых излучателей, расположенных на одной плате с микроконтроллером, без увеличения потребляемого тока, с целью расширения угла обзора сенсора. Разработаны алгоритм работы, программа и макет сенсора с тремя сенсорами и тремя звуковыми излучателями. Алгоритм работы микроконтроллера для устройства с тремя сенсорами представлена на рис. 4.



Рисунок 4 – Алгоритм работы микроконтроллера с тремя сенсорами

Програма базується на основному фрагменті (рис. 5)

```
void UltrasonicSensor: BeepersActivation()
{
    if(echoLengthFirstSensor <= echoBackFront && echoLengthSecondSensor <= echoBackFront &&
        echoLengthThirdSensor <= echoBackFront)
    {
        distanceFirstObstacle = echoLengthFirstSensor / soundSpeed;
        distanceSecondObstacle = echoLengthSecondSensor / soundSpeed;
        distanceThirdObstacle = echoLengthThirdSensor / soundSpeed;
        if(distanceFirstObstacle <= distanceOneMeter &&
            distanceSecondObstacle <= distanceOneMeter &&
            distanceThirdObstacle <= distanceOneMeter)
        {
            if((distanceFirstObstacle < distanceSecondObstacle &&
                distanceSecondObstacle < distanceThirdObstacle)
            {
                /* Determine the number of beeps */
                uint32_t tempFirst = (echoLengthThirdSensor * delayAfterBeep +
                    delayBetweenBeep) /
                    (echoLengthFirstSensor * delayAfterBeep + delayBetweenBeep);
                uint32_t tempSecond = (echoLengthThirdSensor * delayAfterBeep +
                    delayBetweenBeep) /
                    (echoLengthSecondSensor * delayAfterBeep + delayBetweenBeep);
                /* Activating the beeper */
                GPIO_WriteBit(GPIOC, GPIO_Pin_13, Bit_SET);
                Delay(delayBetweenBeep);
                GPIO_WriteBit(GPIOC, GPIO_Pin_13, Bit_RESET);
                for(uint32_t i = 0; i < tempSecond+1; ++i)
                {
                    /* Activating the beeper */
                    GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_SET);
                    Delay(delayBetweenBeep);
                    GPIO_WriteBit(GPIOC, GPIO_Pin_11, Bit_RESET);
                    Delay(echoLengthSecondSensor * delayAfterBeep);
                }
                for(uint32_t i = 0; i < tempFirst+1; ++i)
                {
                    /* Activating the beeper */
                    GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_SET);
                    Delay(delayBetweenBeep);
                    GPIO_WriteBit(GPIOC, GPIO_Pin_8, Bit_RESET);
                    Delay(echoLengthFirstSensor * delayAfterBeep);
                }
            }
        }
    }
}
```

Рисунок 5 – Основной фрагмент программной реализации работы микроконтроллера с тремя сенсорами

Проведен анализ зависимости потребляемого тока от тактовой частоты микроконтроллера и выбрана оптимальная тактовая частота 1 МГц, при этом ток потребления снижается с 3,54 мА на частоте 16 МГц до 0,54 мА на частоте 1 МГц.

В результате испытаний установлено, что суммарный ток потребления устройства составляет 17,7 мА. При использовании литий–полимерного аккумулятора емкостью 2800 мАч, время работы в автономном режиме без подзарядки элемента питания составляет 80 часов, т.е. не менее 10 дней, при этом обеспечивается уверенная фиксация препятствий в диапазоне расстояний от 0,02 до 3 м в пределах диаграммы направленности ультразвуковых датчиков.

Испытания подтвердили, что энергопотребление сонара по сравнению с аналогами [4–10] снижено, а время работы без подзарядки элемента питания увеличено в 4–5 раз.

Наличие двух и трех ультразвуковых датчиков позволяет расширить угол обзора и обеспечить панорамность «зрения», т.е. возможность определять наличие препятствий прямо перед человеком, а также слева и справа от него.

Выводы. На основе проведенных исследований можно сделать следующие выводы:

1. Проведен анализ устройств ультразвукового обнаружения препятствий для слепых.
2. На основании сравнительного анализа, был выбран микроконтроллер с низким энергопотреблением STM8L и оптимальная тактовая частота микроконтроллера.
3. Разработан алгоритм работы микроконтроллера STM8L и программа работы на языке C++. Приведены результаты разработки программной модели и кода программы.
4. Собраны и испытаны макеты сонаров с двумя и тремя сенсорами.
5. Показана возможность наращивания числа сенсоров.
6. Расширен угол обзора сонара за счет наличия нескольких сенсоров.
7. Увеличено время работы устройства в автономном режиме без подзарядки элемента питания за счет применения современной элементной базы, в том числе микроконтроллеров с минимальным энергопотреблением, и разработки соответствующего программного обеспечения.

1. Ершов, С. О. Системы технического зрения с невизуальным представлением пространственной информации / С. О. Ершов и П. Майер // *Известия вузов. Приборостроение*, Т. 51, № 1. С. 42–47, 2008.
2. Ершов, С.О. Локальные и глобальные программно-аппаратные средства позиционирования и навигации для слепых / С.О. Ершов, П. Мейер, Е. Л. Баки-Бородов // *Информатизация и связь*, № 1, с. 58–62, 2011.
3. Jayant, Sakhardande. Arduino Based Mobility Canel / Jayant Sakhardande, Pratik Pattanayak, Mita Bhowmick // *International Journal of Scientific & Engineering Research*, Vol. 4, Issue 4, pp 1163–1166, (April 2013).
4. Бюджетное устройство на базе Arduino для незрячих [Электронный ресурс] / Режим доступа: <https://future.d3.ru/biudzhethoe-ustroistvo-na-baze-arduino-dlia-nezriachikh-open-hardware-750859>.
5. Ультразвуковые фонарики «Сонар-5УФ» [Электронный ресурс] / Режим доступа: <https://www.sites.google.com/site/tehniceskiesredstvadlepyh/1-ultrazvukovoj-fonar-sonar-5uf>.
6. Сонар для слепых на базе Arduino Mini Pro [Электронный ресурс] / Режим доступа: <http://meandr.org/archives/21701>.
7. Shubham, Adhe. Ultrasonic Smart Stick for Visually Impaired People / Shubham Adhe, Sachin Kunthewad, Preetam Shinde, Mrs.V.S.Kulkarni // *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)* e-ISSN: 2278-2834,p- ISSN: 2278-8735.PP 11–15 www.iosrjournals.org.
8. Osama, Bader. 3D Ultrasonic Stick for Blind” / Osama Bader, AL-Barrm, Jeen Vinouth // *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, Vol.3, No.3, pp. 108–114, January 2014.
9. Энергопотребление Arduino [Электронный ресурс] / Режим доступа: https://www.arduino.md/arduino_power_consumption.
10. Sylvain, Cardin. Wearable Obstacle Detection System for visually impaired peopl / Sylvain Cardin, Daniel Thalmann and Frederic Vexo // *Virtual Reality Laboratory (VRlab)Ecole Polytechnique Fédérale de Lausanne (EPFL)CH-1015 Lausanne, Switzerland{sylvain.cardin, daniel.thalmann, frederic.vexo} @epfl.ch*.
11. Костенко, В. Л. Информационно-измерительная система контроля нормированных параметров производственных факторов / Костенко В. Л., Николенко А. А., Поперека Е. Д., Ядрова М. В., Тыманюк К. С. // *Східно-європейський журнал передових технологій. Серія «Інформаційно-управляючі системи»* Т. 3, № 9 (69), С. 51–56, 2014.
12. Костенко, В. Л. Информационная система измерения и контроля параметров воздушной среды / Костенко В. Л., Кондратьев С. Б., Ядрова М. В., Попов Д. А. // *Вісник НТУ «ХПІ». Серія: «Нові рішення в сучасних технологіях»*. – Харків НТУ ХПІ, 2017 – № 23 (1245). – С. 117–122.

УДК 004.491.42

Савенко О.С., к.т.н., Грибинчук В.І., Кульчицький М.О.
Хмельницький національний університет

АРХІТЕКТУРА РОЗПОДІЛЕНОЇ БАГАТОРІВНЕВОЇ ПРОГРАМНОЇ СИСТЕМИ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ЛОКАЛЬНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ

Савенко О.С., Грибинчук В.І., Кульчицький М.О. Архітектура багаторівневої програмної системи виявлення шкідливого програмного забезпечення в локальних комп'ютерних мережах. У статті запропоновано архітектуру розподіленої багаторівневої програмної системи виявлення шкідливого програмного забезпечення в локальних комп'ютерних мережах, особливістю якої є синтез в ній вимог розподіленості, децентралізованості та багаторівневості, що дозволяє її використання автономно. Крім того, особливістю автономних програмних модулів системи є така самоорганізація, що дає змогу здійснювати обмін знаннями в середині системи, яка на відміну від відомих систем дозволяє використовувати знання отримані окремими частинами системи в інших частинах.

Ключові слова: архітектура, розподілена система, шкідливе програмне забезпечення, локальні комп'ютерні мережі.

Савенко О.С., Грибинчук В.І., Кульчицький М.А. Архитектура многоуровневой программной системы обнаружения вредоносного программного обеспечения в локальных компьютерных сетях. В статье предложена архитектура распределенной многоуровневой программной системы обнаружения вредоносного программного обеспечения в локальных компьютерных сетях, особенностью которой является синтез в ней требуемый распределенности, децентрализованности и многоуровневости, что позволяет ее использование автономно. Кроме того, особенностью автономных программных модулей системы есть такая самоорганизация, что позволяет осуществлять обмен знаниями внутри системы, которая в отличие от известных систем позволяет использовать знания, полученные отдельными частями системы в других частях.

Ключевые слова: архитектура, распределенная система, вредоносное программное обеспечение, локальные компьютерные сети.

Savenko O.S., PhD, Gribinchuk V.I., Kulchytsky M.O. The architecture of the multi-level software system for detecting malware in local computer networks. The article proposes the architecture of distributed multilevel software system for detection of malicious software in local computer networks, the feature of which is the synthesis of its requirements for distribution, decentralization and multilevel, allowing its use autonomously. In addition, the feature of autonomous program modules of the system is the same organization, which allows the exchange of knowledge in the middle of the system, which, unlike the known systems, allows you to use the knowledge gained by separate parts of the system in other parts.

Keywords: architecture, distributed system, malicious software, local computer networks.

Постановка проблеми. На сьогодні шкідливе програмне забезпечення (ШПЗ) представляє собою складні багатофункційні програмні системи та комплекси, які побудовані з використанням ефективних методів створення програмних засобів та методів поширення зловмисного коду. При цьому воно переважно розроблено для використання в комп'ютерних мережах (наприклад, ботнет). Для організації ефективної протидії таким засобам важливим є розробка таких систем виявлення ШПЗ, архітектура яких враховувала б ці особливості. Крім того, досягнення підвищення достовірності виявлення ШПЗ в межах тільки однієї комп'ютерної системи (КС), яка має вихід в мережу Internet, може бути недостатнім при протидії засобам ШПЗ, які представлені великим програмним комплексом, що розміщений в багатьох комп'ютерних системах в глобальній мережі, і частини якого комунікують між собою.

Аналіз та тестування відомих програмних засобів виявлення нового шкідливого програмного забезпечення, відомості про яке не містяться в антивірусних засобах, вказують на недостатньо високий рівень достовірності виявлення [1]. Реалізація нових методів виявлення конкретних типів шкідливого програмного забезпечення шляхом створення відповідних інформаційних систем потребує подальшого розвитку з метою підвищення достовірності виявлення ШПЗ.

Аналіз досліджень і публікацій. Проблемі виявлення ШПЗ сьогодні приділяється значна увага. Для виявлення та протидії ШПЗ розроблено ряд систем та методів.

В роботі [2] запропоновано систему виявлення кібератак на основі залучення нейромережних імунних детекторів. Розроблена система складається з двох частин. Перша реалізована апаратно й працює постійно в режимі реального часу. Друга частина представлена програмним забезпеченням на виділеному комп'ютері, який використовується для аналізу поточних атак та створення відповідних засобів захисту. Прийняття рішення про можливий вплив

ШПЗ здійснюється із залученням системи нейромережних детекторів, в основу якої закладено алгоритм Мамдані.

В роботі [3] представлено інтелектуальну адаптивну систему виявлення ШПЗ на основі інтеграції штучних імунних систем та штучних нейронних мереж. Така система працює за основними принципами штучної імунної системи, де імунні детектори представляють нейронну мережу і виявляють шкідливий шаблон за допомогою аналізу структури виконуваного коду. Запропонована система володіє функціями самоадаптивності та самонавчання та дозволяє здійснювати виявлення як відомих, так і нових видів ШПЗ.

Авторами роботи [4] запропоновано систему ідентифікації та класифікації для мережних кібератак. Для реалізації системи запропоновано використання комбінації різних методів штучного машинного навчання, а саме нейронних мереж, імунної системи, нейрофізичних класифікаторів та метод опорних векторів. Відмінною особливістю запропонованої системи є багаторівневий аналіз мережевого трафіку, що дає можливість виявляти атаки методом підпису та комбінувати набір адаптивних детекторів на основі методів машинного навчання.

У роботі [5] запропоновано гібридну систему виявлення рідковживаних кібератак на основі використання штучних імунних систем та нечіткої кластеризації (FC-ANN). Спочатку FC-ANN розподіляє навчальні дані на декілька підгруп з використанням методів нечіткої кластеризації. Для отримання остаточних результатів, визначається оцінка для кожного елементу з сформованих підгруп та виконується їх поєднання з використанням штучної імунної системи.

Проведений аналіз показав, що для виявлення ШПЗ відомі системи здійснюють аналіз мережного трафіку, файлів аудиту, пакетів, що передаються по мережі, перевіряють конфігурацію відкритих мережевих сервісів. Для встановлення факту порушення роботи КС, відомі системи використовуються різні методи машинного навчання, а саме нейронні мережі, штучні імунні системи, метод опорних векторів, Байєсові мережі, нечітка кластеризація. Проте, основним недоліком представлених систем є їх хост-орієнтований підхід до виявлення ШПЗ.

Для того, щоб ефективно застосовувати методи та засоби виявлення шкідливого програмного забезпечення необхідно розробити систему, яка б включала в себе достатню кількість реалізованих ефективних методів у вигляді відповідних підсистем, мала можливість до нарощування та враховувала б майбутні тенденції розвитку як антивірусних засобів, так і шкідливого програмного забезпечення.

Виклад основного матеріалу дослідження. Виявлення шкідливого програмного забезпечення, що здійснюється за допомогою різноманітних засобів, суттєво залежать, зокрема, і від архітектури таких засобів, а також їх позиціонування та місця розміщення в комп'ютерних системах локальних мереж. Враховуючи, що процес виявлення ШПЗ проводитиметься в локальних мережах, то вибір моделі функціонування системи повинен передбачати залучення інформації зі всіх комп'ютерних систем локальної мережі, тобто розміщення на всіх КС цієї системи. Це необхідно для підвищення ефективності і достовірності виявлення за рахунок врахування інформації про стан з інших КС для прийняття рішення на конкретній КС. Ці основні вимоги, що система повинна бути розміщена в мережі на кожній КС, впливають на вибір моделі її архітектури. Також, важливим для таких систем є міграція центру прийняття рішень системи в локальній мережі, оскільки атака на нього призведе до виведення всієї системи з робочого стану. Система повинна бути побудована так, щоб розміщені в КС локальної мережі її частини ефективно взаємодіяли між собою для обміну інформацією про стан КС з метою надання додаткової інформації для прийняття рішення. Крім того, система виявлення ШПЗ повинна відповідним чином структуруватись, щоб мати можливість нарощення, причому її збільшення не повинно сповільнювати процес виявлення. Основними функціями такої розподіленої багаторівневої програмної системи (РБПС) є перевірка наявного програмного забезпечення та запуску процесів в КС локальної мережі на можливість віднесення до шкідливого програмного забезпечення. Досягнення відповідності системи заданим характеристикам та покладеним на неї функціям з виявлення ШПЗ формують вимоги до неї, основними з яких будуть такі: розподіленість, децентралізованість, самостійність у прийнятті рішень, багаторівневність, самоорганізованість, адаптивність. Врахування таких складових в моделі системи є основою її архітектури і підвищить надійність функціонування та живучість системи в локальній мережі. Беручи до уваги умову, що виявлення шкідливого програмного забезпечення відбуватиметься у локальній мережі, що складається з комп'ютерних систем, представимо схему структурної розподіленості архітектури системи, яка відображена на рис.1.

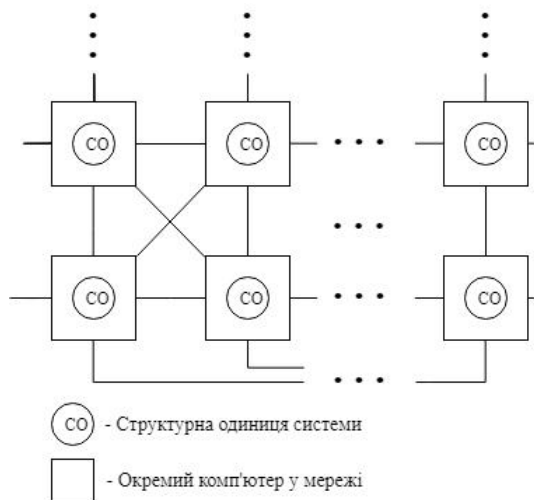


Рис.1 Схема структурної розподіленості архітектури

Кожна одиниця системи на окремих комп'ютерах в мережі є автономною та повноцінною, з'єднаною з кожною іншою. Таким чином жодна з них не виступає у ролі центрального керівного пункту прийняття рішень. Система здатна розширюватися підсистемами, що можуть бути приєднані до основної архітектури. Оскільки структура системи є просторово розподіленою, а кожна одиниця системи здатна приймати самостійні рішення, тому для позначення кожної частини системи будемо використовувати поняття автономного програмного модуля (АПМ) системи. Враховуючи децентралізованість та автономність системи, механізм взаємодії окремих АПМ базується на наступних принципах: АПМ виконуються заплановані завдання без впливу ззовні, кожен АПМ може обмінюватися з кожним іншим АПМ, система здійснює самоконтроль та планує власні цілі.

Дотримання цих принципів при реалізації механізму взаємодії окремих модулів значно впливатиме на ефективність досягнення поставлених перед системою цілей.

Згідно зображеної вище схеми структурної розподіленості системи у мережі, вона представлена однаковими одиницями, що знаходяться на кожному окремому комп'ютері у мережі. Запущені АПМ в мережі формують розподілену систему. За такої схеми система здатна виконувати свої задачі, навіть якщо активними під час її роботи є лише декілька комп'ютерів із встановленими на них АПМ. Тобто система складається з тих АПМ, які є активними.

Децентралізованість архітектури представлено на узагальненій структурній схемі складових системи на рис.2.

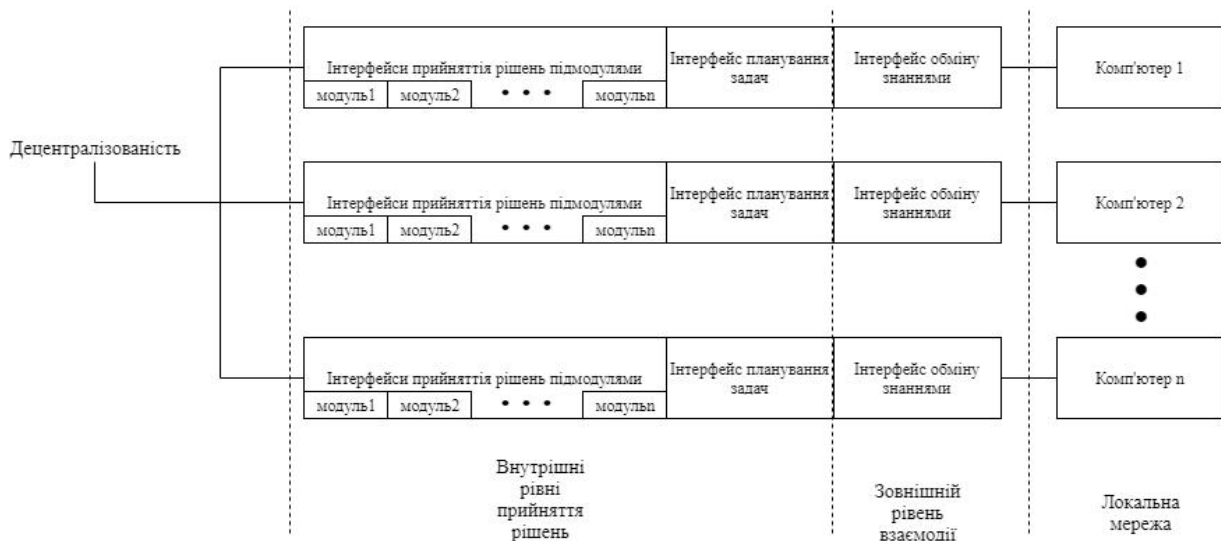


Рис. 2. Узагальнена схема складових системи

Узагальнена схема складових системи відображає реалізовану ідеологію децентралізованості та розподіленості у просторі. Прийняття рішень не здійснюється єдиним центром, проте виконується на внутрішніх рівнях через інтерфейси прийняття рішень. Кожна окрема одиниця є автономною у прийнятті рішень.

Автономність кожної одиниці на внутрішньому рівні включає здатність прийняття рішення про виявлення спроби атаки шкідливим програмним забезпеченням та передачу повідомлення на зовнішній рівень.

На зовнішньому рівні здійснюється обмін знаннями з іншими одиницями, згідно чого система здатна реагувати на спроби атаки, накопичувати «досвід» та адаптуватися до середовища, в якому вона працює, що забезпечує гнучкість системи до зміни обставин.

Таким чином система, в цілому, володіє властивістю адаптуватися для підвищення ефективності виявлення та реагування на атаки шкідливого програмного забезпечення. Ця властивість включає в себе гнучку зміну функціонування підмодулів для досягнення оптимальної роботи в середовищі.

Кожний АПМ у розподіленій системі ідентифікується по системним характеристикам комп'ютерної системи, на яку він встановлений та її IP-адреси у мережі. З цією метою при першому запуску АПМ на комп'ютері виконується сканування інформації про апаратні характеристики системи.

Абстракція децентралізованості включає в себе здатність кожної одиниці системи на окремому комп'ютері запустити певний функціонал виявлення шкідливого програмного забезпечення, виконати перехід між різними рівнями на основі інформації отриманої з іншого рівня, провести обмін повідомленнями з іншими одиницями всієї системи, отримати знання від інших частин та передати їх на інтерфейси прийняття рішень.

Таким чином, на внутрішньому рівні прийняття рішень відбуваються основні задачі виявлення шкідливого програмного забезпечення, в той час як на зовнішньому рівні відбувається обмін результатами та прийняття більш глобальних рішень стосовно загальної організації роботи системи у мережі. Зовнішній рівень є каналом взаємодії та відповідає за спілкування між активними АПМ системи. За його допомогою є можливою комунікація між окремими АПМ розподіленої системи.

Взаємодія внутрішнього рівня прийняття рішень та зовнішнього рівня взаємодії представлено на рис. 3.



Рис.3. Схема взаємодії внутрішнього та зовнішнього рівнів

Робота кожного АПМ системи розпочинається із запуску комп'ютера в мережі, на якому він інстальований, та запуску цього модуля. Після запуску модуля виконується перехід до планування задач модуля. Відповідно до запланованої задачі відбувається активізація окремого підмодуля та запуск його функціоналу. Результати функціонування цього підмодуля надсилаються іншим частинами системи через зовнішній рівень взаємодії. Це дозволяє іншим АПМ системи бути сповіщеними про поточний стан кожної іншої частини системи.

У разі виявлення певним підмодулем атаки шкідливого програмного забезпечення активізується механізм реагування системи на загрозу. Цей механізм включає в себе процедуру відправки критичного повідомлення на усі інші АПМ системи. Якщо АПМ знаходиться у

критичному стані приймається рішення про завершення роботи модуля. Описана взаємодія між внутрішнім та зовнішнім рівнями відображена на рис. 4.

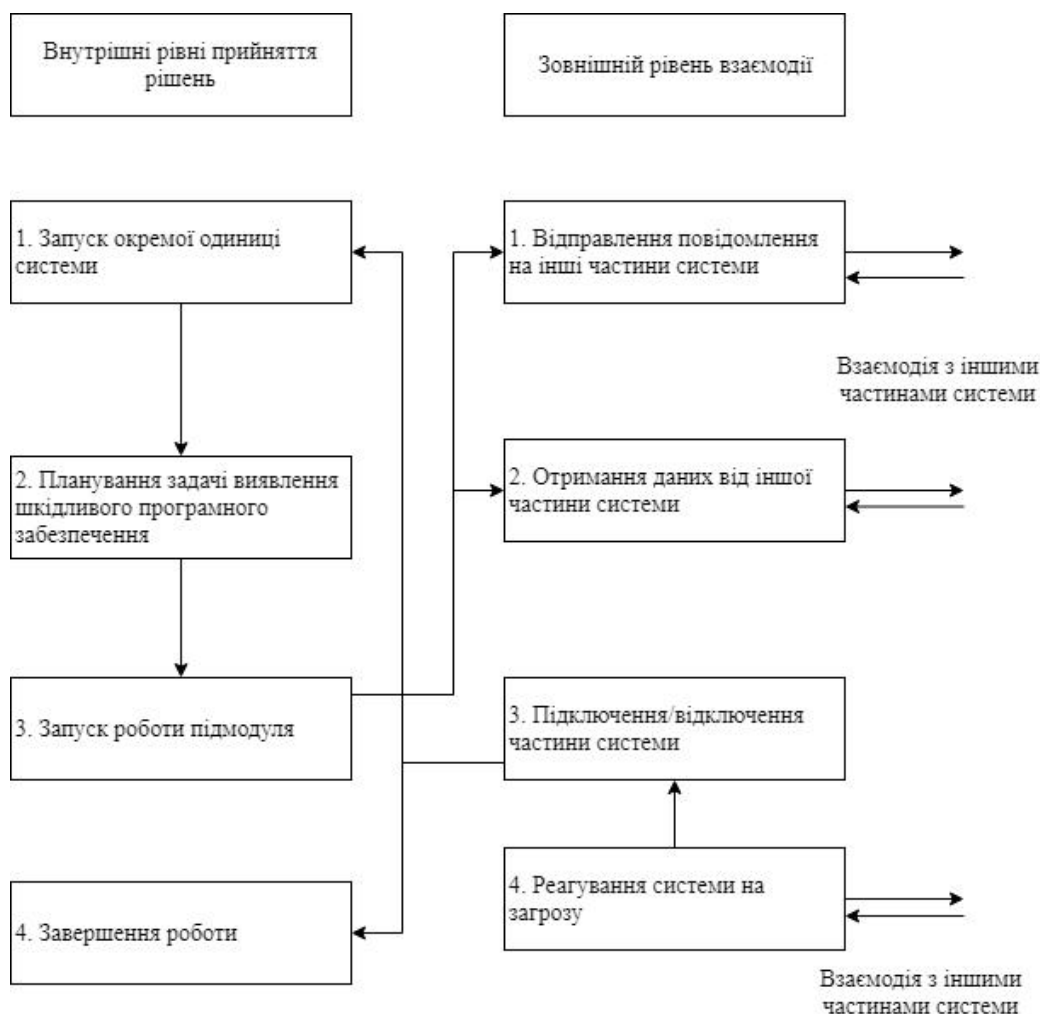


Рис. 4. Взаємодія між внутрішнім та зовнішнім рівнями

Частини системи взаємодіють між собою за допомогою повідомлень, причому кожне з них складається із заголовку та тіла.

В якості заголовку повідомлення виступає АПМ-ініціатор повідомлення – ідентифікатор АПМ, який надіслав повідомлення. Як ідентифікатор АПМ використовується IP-адреса комп'ютерної системи, на якій цей модуль встановлений.

Тіло повідомлення містить два поля:

- 1) тип повідомлення – ідентифікатор типу повідомлення;
- 2) дані – безпосередньо дані повідомлення.

Тип повідомлення використовується для отримання інформації про дані, що міститимуться безпосередньо у тілі повідомлення.

Можливі типи повідомлень представлено в таблиці 1. Залежно від типу повідомлення його тіло міститиме різні дані.

Таблиця 1. Типи повідомлень

№	Тип повідомлення	Опис типу
1	Модуль активовано	Повідомлення, що надсилається усім частинами системи у випадку активації модуля на конкретній комп'ютерній системі. Виступає у якості прохання відгукнутися усіх активних модулів у системі.
2	Модуль деактивовано	Повідомлення, що надсилається усім частинами системи у випадку деактивації модуля на конкретній комп'ютерній системі
3	Повідомлення-вітання	Повідомлення, що надсилається тому модулю, який запросив відгукнутися усіх інших активних частинам системи, з метою виявлення усіх активних частин системи.
4	Зміна задачі	Повідомлення, що надсилається усім частинами системи у випадку зміни виконуваної задачі конкретним АПМ.
5	Задача виконана	Повідомлення, що надсилається усім частинами системи у випадку завершення роботи над виконуваною задачею конкретним АПМ.
6	Повідомлення-опитування	Повідомлення, що надсилається усім частинами системи з метою отримання інформації про поточний стан кожної частини.
7	Повідомлення-стан	Повідомлення, що надсилається тому модулю, який запросив відгукнутися усіх інших активних частинам системи, з метою отримання інформації про поточний стан кожної частини.
8	Виявлено загрозу	Повідомлення, що надсилається усім частинами системи з метою інформування про виявлення загрози.

Стани, в яких перебувають АПМ змінюються під час їх роботи. Уся інформація стосовно зміни робочого стану надсилається усім іншим активним у мережі АПМ через повідомлення. Спеціальний підмодуль виконує обмін повідомленнями з іншими частинами системи. Якщо жоден інший АПМ у системі не активований, відповіді на розіслане повідомлення отримано не буде. Час відправки та отримання кожного повідомлення зафіксовується. Обмін повідомленнями відбувається усіма АПМ з усіма іншими активними АПМ. Кожний АПМ зберігає у локальній базі даних статистичні дані стосовно власної роботи та інформацію стосовно результатів роботи інших частин системи. Така збережена інформація може бути використана для гнучкої адаптації, тобто автоматичної корекції роботи системи до обставин у середовищі, адже кожен модуль є поінформованим стосовно інших частин системи.

Залежно від кількості активних АПМ системи у мережі та від інтенсивності завантаження кожного конкретного модуля, за певну одиницю часу у базу даних може бути записана велика кількість інформації. Причому значний відсоток записаної інформації становитиме чисто технічні записи. Наприклад, повідомлення про те, що у певний момент часу АПМ активізувався на певній комп'ютерній системі. Звісно, через певний період часу такого роду інформація може бути вилучена з бази. Тому з метою оптимізації знань, накопичених у базі даних, передбачено підмодуль, функціонал якого забезпечує здійснення вилучення неактуальних для системи даних. В результаті накопичення досить великого об'єму знань, це дозволяє покращити ефективність роботи системи в цілому.

Кожен запис у базі даних має спеціальну мітку, залежно від типу даних, які він представляє:

- 1) звичайне "технічне" повідомлення;
- 2) попередження;
- 3) помилка у роботі якогось підмодуля АПМ;
- 4) критичний стан АПМ;
- 5) повідомлення про результат виконання завдання.

Усі АПМ у системі є структурно ідентичними та складаються з набору підмодулів, кожен з яких виконує певну задачу. Узагальнена схема робочих станів, в яких може перебувати кожен АПМ в процесі своєї роботи представлена на рис. 5 у вигляді графу.

Отже, можна виділити 9 основних станів, у яких знаходитиметься АПМ певної комп'ютерної системи у конкретний момент часу:

- 1) активізація автономного програмного модуля;

- 2) планування наступної задачі на виконання, визначення переходу до станів 3, 4, 5, 6, 7, 8;
- 3) перевірка файлів на жорсткому диску, визначення переходу до станів 2, 5;
- 4) перевірка запущених процесів в оперативній пам'яті, визначення переходу до станів 2, 6;
- 5) сканування виконуваних файлів на жорсткому диску, перехід до стану 2;
- 6) сканування запущених процесів, перехід до стану 2;
- 7) виконання оптимізації записів у базі даних, перехід до стану 2;
- 8) завершення роботи автономного програмного модуля;
- 9) позначення комунікації з іншими модулями у системі, формально АПМ не переходить до цього стану.

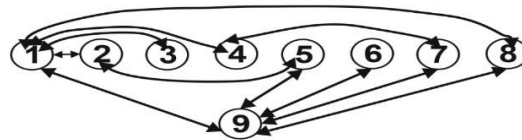


Рис. 5. Узагальнена схема робочих станів АПМ системи у вигляді графу

АПМ потрапляє у стан 1 при його активізації або користувачем, або комп'ютерною системою, на яку він встановлений, що спричиняє процес перевірки характеристик комп'ютерної системи та порівняння їх із збереженими раніше характеристиками. Після чого відбувається формування спеціального повідомлення про активацію роботи іншим АПМ у системі.

У стан 8 АПМ потрапляє при деактивації або користувачем, або при завершенні роботи комп'ютерної системи, на яку він встановлений. Деактивація АПМ відбувається лише після того, як усі інші активні АПМ будуть поінформовані спеціальним повідомленням про вимкнення. Основним робочим станом є 2 стан, на якому виконуються збір результатів виконання попередньої задачі та планування наступної. Залежно від результатів моніторингу подій у системі або змін на конкретній комп'ютерній системі відбувається прийняття рішення про перехід на один із станів - 3, 4, 5, 6 та 7, - на яких АПМ зайнята конкретною задачею. Стани 3 та 5 позначають процес перевірки та сканування файлів на жорсткому диску комп'ютерної системи, а 4 та 6 - перевірка та сканування запущених процесів. Перебуваючи в 7 стані, АПМ здійснює оптимізацію записів у локальній базі даних.

Стан 9 позначає процес обміну отриманими результатами з іншими активними частинами системи. Переходячи з одного стану в інший АПМ формує спеціальне повідомлення з інформацією про перехід до виконання певного завдання. Після завершення завдання АПМ розсилає повідомлення з отриманими результатами. Обробка отриманих результатів на всіх АПМ виконується за єдиним алгоритмом, після чого відбувається реагування на отриманий результат. Кожний АПМ позначено як "Defender Application" за назвою реалізованої системи. Перелік функціональних можливостей, якими володіє розроблена система з погляду користувача цієї системи відображено на use-case діаграмі на рис. 7.

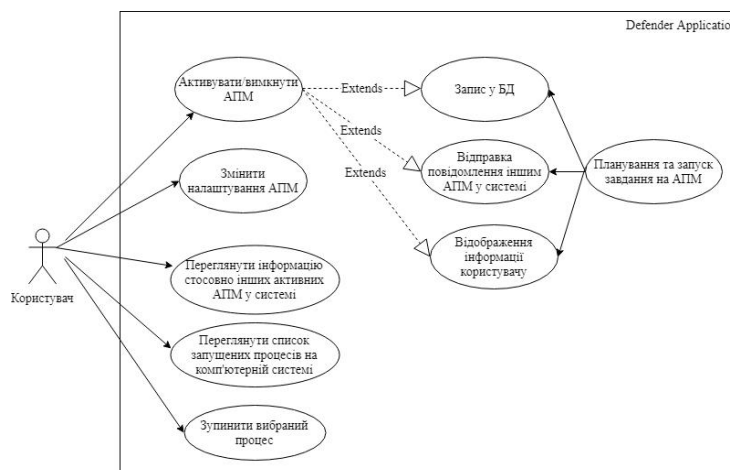


Рис. 7. Use-case діаграма розробленої системи

Взаємодія між користувачем та розробленою системою відображена на рисунку 3.9 у вигляді діаграми послідовностей.

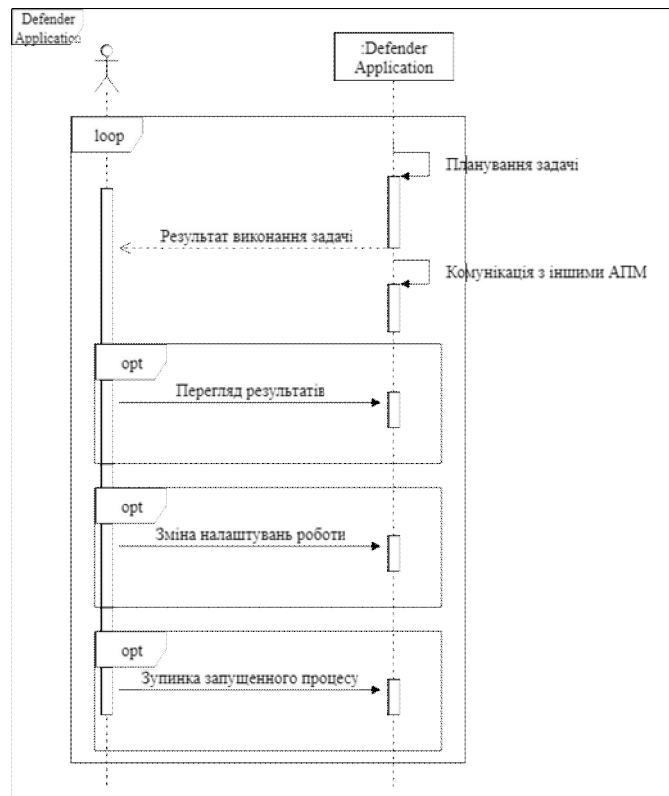


Рис. 8. Діаграма послідовностей розробленої системи

Як видно із наведеної вище діаграми послідовностей, кожен окремий модуль системи встановлений на конкретній комп'ютерній системі є автономним, тобто не потребує впливу ззовні для прийняття рішень стосовно планування завдань та обміну повідомленнями з іншими модулями у системі. Користувач може лише переглядати результати, змінювати деякі налаштування та використовувати додатковий функціонал, наприклад, зупиняти запущені процеси. Головне вікно користувацького інтерфейсу відображено на рис. 9.

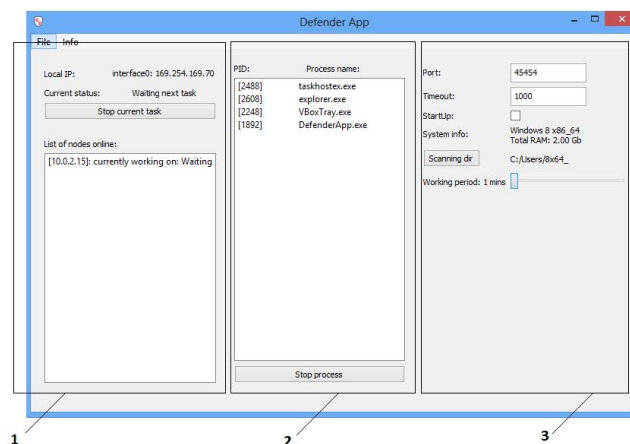


Рис. 9. Головне вікно користувацького інтерфейсу Defender App

Як видно із рисунку 9, користувацький інтерфейс розділено на три основні області:

- 1) інформація про поточний стан роботи та список інших модулів системи, які є активними на даний момент у мережі;
- 2) список запущених процесів на даній комп'ютерній системі;
- 3) налаштування.

Висновки. Розроблена архітектура розподіленої багаторівневої програмної системи базується на принципах децентралізації та самоорганізації і дозволяє здійснювати її наповнення різними функціоналами виявлення шкідливого програмного забезпечення в локальних комп'ютерних мережах. РБПС відноситься до реагуючих систем, яка постійно здійснюватиме моніторинг запущених процесів та виконуваних програм в комп'ютерних системах мережі. Об'єктами для дослідження зі сторони РБПС є перевірка наявного програмного забезпечення та запущених процесів в КС локальної мережі на можливість віднесення до шкідливого програмного забезпечення. Основою архітектури РБПС виступають автономні програмні модулі з однаковими архітектурами, але при цьому кожен з них може самостійно приймати рішення на основі різних даних зібраних з різних КС мережі. Для ефективної роботи РБПС необхідним є розробка методів і моделей взаємодії та узгодження роботи різних програмних модулів між собою та їх відповідних рівнях і деталізація структури її станів.

1. Virus Bulletin URL: <https://www.virusbulletin.com/testing/> (дата звернення: 25.03.2018).
2. High performance adaptive system for cyber attacks detection / Komar M. et al. *The 9-th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications: Proceedings* (Bucharest, 21-23 September 2017). Bucharest, 2017. P. 853–858.
3. Golovko V., Bezobrazov S. Neural Network Artificial Immune System for Malicious Code Detection. *Brest State Technical University*. 2015, P. 1–7.
4. Branitskiy A., Kotenko I. Hybridization of computational intelligence methods for attack detection in computer networks. *Journal of Computational Science*. 2017. No.23 P. 145–156.
5. A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering / Wang G., Hao J., Ma J., Huang L. *Expert Systems with Applications: An International Journal*. 2010. Vol. 37. Issue 9. P. 6225–6232.

УДК 004.65

Струзік В.А., Грибков С.В., Литвин А.О.

Національний університет харчових технологій

ДОСЛІДЖЕННЯ МЕТОДІВ І ПІДХОДІВ ПРОВЕДЕННЯ РЕФАКТОРІНГУ БАЗ ДАНИХ

Струзік В.А., Грибков С. В., Литвин А. О. Дослідження методів і підходів проведення рефакторінгу баз даних. В роботі проведено дослідження методів і підходів проведення рефакторінгу баз даних. Досліджено основні ознаки при яких необхідно проводити рефакторинг баз даних, а також розглянуто основні операції рефакторінгу.

Ключові слова: рефакторинг, ітераційні процеси, розробка, інформаційні системи.

Струзик В.А., Грибков С. В., Литвин А. О. Исследование методов и подходов проведения рефакторинга баз данных. В работе проведено исследование методов и подходов проведения рефакторинга баз данных. Исследовано основные показатели при которых необходимо проводить рефакторинг баз данных, а также рассмотрено основные операции рефакторингу.

Ключевые слова: рефакторинг, итерационные процессы, разработка, информационные системы.

Struzik V.A., Hrybkov S.V., Lytvyn A.O. Study of the methods and approaches of databases refactoring In this paper the research of methods and approaches of databases refactoring is carried out. The main features under which it is necessary to refactor databases are examined, as well as the basic operations of refactoring.

Keywords: refactoring, databases, information systems.

Актуальність. Розвиток інформаційних технологій спонукає до постійного розвитку та удосконалення методологій по створенню інформаційних систем. Постійне удосконалення методологій спрямовано не тільки для покращення показників програмного забезпечення, а й на зменшення часу розробки, і головне – забезпечення якості менеджменту процесу створення програмного забезпечення. Всі сучасні методології розробки програмного забезпечення за своїм характером є еволюційними, тому для них характерне постійний ітеративний процес на якому постійно відбувається накопичення та вдосконалення функціоналу. До прикладів типових процесів відносяться: Rational Unified Process (RUP); Extreme Programming (XP); Scrum; Dynamic System Development Method (DSDM); сімейство методів Crystal; Team Software Process (TSP); Agile Unified Process (AUP); Enterprise Unified Process (EUP); Feature-Driven Development (FDD); Rapid Application Development (RAD). На сьогоднішній день, незалежно від життєвого циклу програмного забезпечення етап розробки програмного забезпечення є ітеративним, що забезпечує: гнучке підпорядкування вимог клієнтів; швидке розширення функціоналу програмного забезпечення; просту взаємодію розробників; зменшення .

Популярність ітеративного підходу розроблення програмного забезпечення обумовлено його ефективністю, а саме: врахування усіх вимог замовника, що формуються послідовно; швидку зміну функціоналу системи; ефективну взаємодію між елементами програмного забезпечення; продуктивну роботу розробників без додаткових витрат ресурсів та з мінімізацією часу на розробку. Але дана методологія потребує високого рівня комунікації між членами команди розробників. Цей процес відрізняється від послідовного підходу, в якому виявляються всі вимоги, що підлягають реалізації – створюється детальний проект, який реалізується, виконується його перевірка і, нарешті, відбувається розгортання готової системи.

Слід зазначити, що більшість компаній на практиці не завжди дотримуються однієї чіткої методології. Але виникають ситуації, коли замовник висуває вимоги до дотримання певної методології, тому для спеціалістів, які працюють в ІТ сфері необхідно знати основні методології по створенню програмного забезпечення. Все це обумовлено тим, що замовники висувають вимоги по сертифікації відповідно до конкретної методології.

Постійний розвиток інформаційних систем спонукає до постійного саморозвитку. Чітке розуміння та знання методологій дозволяє краще розуміти процес створення програмного забезпечення та організацію на різних стадіях його життєвого циклу. Методології частково перетинаються у різних автоматизованих засобах по проектуванню та створенню програмного забезпечення.

На різних етапах життєвого циклу програмного забезпечення виникає задача рефакторінгу його елементів та складових з метою підвищення ефективності та якості функціонування кінцевого

продукту, забезпечення сумісності з рішеннями інших розробників. Окрему нішу займає процес рефакторингу баз даних, що потребує детальне вивчення та класифікацію підходів.

Аналіз останніх досліджень і публікацій. Авторами роботи [1] розглядаються стратегії рефакторингу та складності його проведення при створенні інформаційних систем, але більше уваги приділено рефакторингу коду, а не структури бази даних. Робота [2] присвячена еволюційному підходу розробки програмних засобів, а також висвітлено ефективність даного підходу при проведенні рефакторингу. У праці [3] наведено аспекти проведення рефакторингу, але не розглянуто рефакторинг структури баз даних, а тільки окремих елементів. Роботи [4, 5] описують приклади проведення рефакторингу, але більша увага направлена на рефакторинг програмного коду. Автори праці [6] наводять основні алгоритми та підходи проведення рефакторингу усіх елементів системи, описано програмні засоби для підтримки процесу рефакторингу. Більше уваги приділено на оптимізацію процесу рефакторингу потоків даних, а не на структури зберігання даних. Робота [7] більш присвячена рефакторингу баз даних з підтримкою SQL та такі, що мають реляційну структуру. Автори роботи [8] висвітлили проблеми проектування інформаційних систем, а також який вплив має обрання технологій та підходів розробки на проведення рефакторингу.

Виділення невирішених раніше частин загальної проблеми. На даний момент підходи рефакторингу програмного коду описані в більшості фундаментальних літературних джерелах, існують багато підходів та автоматизованих програмних засобів для його проведення. Процес рефакторингу програмного коду в теорії не чим не відрізняється від практичного застосування [1]. Але особливу увагу необхідно виділити рефакторингу баз даних, що направлений на підвищення ефективності роботи з нею із забезпеченням збереження її функціональної та інформаційної семантики.

Формлювання мети дослідження. Необхідно провести чітку класифікацію та дослідити етапи рефакторингу баз даних, що дасть можливість підвищити ефективність його проведення.

Виклад основного матеріалу досліджень. Рефакторингом бази даних називають зміну в схемі бази даних, що сприяє поліпшенню проекту і в той же час забезпечує збереження функціональної та інформаційної семантики бази даних. Іншими словами, проведення операцій рефакторинга не повинно призводити до додавання нових функцій або порушення роботи існуючих, а також не повинно бути направлено на додавання нових даних або зміну існуючих даних. Схема бази даних включає структурні аспекти, такі як визначення таблиць і представлень, а також функціональні аспекти, такі як збережені процедури й тригери.

Але особливу увагу необхідно виділити рефакторингу баз даних, що направлений на підвищення ефективності роботи з нею із забезпеченням збереження її функціональної та інформаційної семантики. Необхідно відмітити, що проведення рефакторингу бази даних залежить від того, на якій стадії знаходиться програмне забезпечення, адже чим ближче воно до стадії впровадження та експлуатації, тим буде складнішим це здійснити, а також буде збільшуватися витрати часу. Рефакторинг баз даних набагато складніший, ніж рефакторинг програмного коду, тому що база даних містить у собі [1–2]:

- структуровані дані, збережені у відповідності з певною схемою;
- програмні конструкції, такі як збережені процедури та тригери, що забезпечують виконання бізнес-логіки.

Рефакторинг збережених програмних конструкцій бази даних здійснюється за стандартними підходами рефакторингу програмного коду та підлягає автоматизованому тестуванню. Але будь-які зміни у базі даних стосуються не тільки форматів та структур збереження інформації в ній, а впливають на підходи та функціонал програмного забезпечення у цілому. Особливу проблему при рефакторингу бази даних створюють наявні дані.

Основними причинами проведення рефакторингу баз даних є наявність наступних недоліків [2]:

- багатоцільові стовпці, що використовуються в декількох цілях та в різних додатках;
- багатоцільові таблиці, що використовуються для зберігання даних про сутності декількох різних типів;
- надлишковість даних, що є суттєвою проблемою в повсякденній експлуатації бази даних, оскільки при зберіганні одних і тих же даних в декількох місцях виникає ймовірність порушення відповідності та узгодженості;

- таблиці з великою кількістю стовпців, що свідчить про наявність полів, які можуть відноситись до різних сутностей;
- багатозначні стовпці, що вміщують в різних позиціях кілька різних фрагментів інформації;
- багатозначні стовпців, всередині яких в різних позиціях представлено кілька різних фрагментів інформації.

Розглянемо детально основні причини для проведення рефакторингу.

Багатоцільові стовпці. Якщо стовець використовується в декількох цілях, то велика ймовірність, що існує додатковий код, який призначений для забезпечення використання вихідних даних за призначенням, шляхом перевірки значень в одному або декількох інших стовпцях.

Багатоцільові таблиці – якщо таблиця використовується для зберігання даних про сутності кількох різних типів, що не можливо допускати для коректної роботи з базою даних. Прикладом може слугувати таблиця Клієнт, яка використовується для зберігання інформації про фізичних осіб та фірми. Недоліком такого підходу є те, що інформації про фізичних осіб і фірми мають різні структури даних, наприклад, для фізичних осіб необхідно зберігати відомості про прізвища, імені та по-батькові, а для фірм вказується юридична назва. За таким випадком неминуча поява значення NULL у певних стовпчиках, що відносяться до одного з типів клієнтів.

Надлишкові дані. Наявність надлишкових даних у базі даних, що експлуатується, неминуче призведе до порушення цілісності, тому що не доцільно в різних місцях зберігати однакову інформацію. Наприклад, у багатьох організаціях інформація про клієнтів зберігається декілька разів з різною уточнюючою інформацією. Наприклад клієнт міг змінити адресу проживання, номер мобільного телефону тощо, а в таблиці буде кількість записів, що мають різні значення певних атрибутів.

Таблиці з великою кількістю стовпців. Якщо в таблиці є багато стовпців, це можна розглядати як ознаку відсутності неподільності в структурі таблиці. В такій таблиці можуть бути представлені дані, що відносяться до кількох різних сутностей, що потребує проведення нормалізації структури.

Таблиці з великою кількістю записів. Наявність великих таблиць слугує ознакою проблеми продуктивності. Наприклад, при пошуку в таблиці, що вміщує понад мільйонну записів, не можливо без втрати часу. У зв'язку з цим, доцільно частину записів, що застарілі чи неактуальні, перенести в архів.

Багатозначні стовпці. Багатозначними називаються стовпці, що поєднують кілька різних фрагментів інформації. Наприклад, багатозначним стовпцем є ідентифікатор клієнта, в якому перші чотири цифри ідентифікатору клієнта позначають головне відділення компанії цього клієнта, оскільки для виявлення додаткової інформації доводиться виконувати синтаксичний аналіз значень з цього стовпчика. На практиці, такі стовпці розбиваються на окремі поля з даними, щоб можна було простіше проводити обробку цих полів у вигляді окремих елементів.

Наявність нереалізованих змін. Якщо зміни в схемі бази даних давно не проводилися, з тієї причини, що можуть виникнути якісь порушення в роботі. Подібні побоювання перед можливими порушеннями в роботі явно свідчать про постійне зростання ризику повної відмови системи, а така ситуація з часом тільки погіршується.

При проведенні рефакторингу бази даних не повинно відбуватися додавання нових функціональних можливостей або порушуватися робота існуючих, а також не повинні додаватися нові дані або змінюватися існуючі, але при цьому можливі зміни як у схемі бази даних так і в модулях забезпечення бізнес-логіки обробки інформації, тому що вони прив'язані до схеми [1].

Також необхідно відмітити, що іноді необхідно провести достатньо глибокий рефакторинг, що буде стосуватися дуже суттєвих змін з метою оптимізації та вдосконалення, при збереженні функціональності програмного забезпечення.

При проведенні рефакторингу баз даних доцільно виділити 6 основних операцій, а саме:

- рефакторинг структури БД;
- рефакторинг якості даних;
- рефакторинг зв'язків та цілісності;
- рефакторинг архітектури;
- рефакторинг методів;
- зміни, що не входять до операцій рефакторингу.

Доцільно розглянути їх детальніше.

Операції рефакторингу структури направлені на зміну структур однієї чи декількох таблиць або представлень. Прикладом такої операції може бути переміщення стовпця з однієї таблиці в іншу або розбиття багатоцільового стовпчика на кілька окремих стовпців, кожен з яких виконує окреме призначення.

Операції рефакторингу направлені на якість даних в першу чергу повинні забезпечити ефективність збереження та роботи з такими даними, що досягається за рахунок додавання правил валідації, форматів введення даних, забезпечення неможливості залишати порожніми поля та використання значень за замовчуванням та ін.

Проведення операції рефакторингу направлених на зв'язки та цілісність даних забезпечують зв'язаність даних, що знаходяться в різних таблицях зв'язаних різними видами зв'язків. При таких операціях доцільним є створення тригерів для: реалізації каскадного типу зв'язку; ведення історії проведених операцій; збереження видалених чи змінених записів у додаткових таблицях та ін.

Операції рефакторингу архітектури направлені на розбиття складних операцій бізнес-логіки, що реалізовані у клієнтських додатках, на простіші та реалізації їх у збережених процедурах та функціях, що дозволяє уніфікувати подальше їх використання у різних програмних модулях та додатках.

Операції рефакторингу методів полягає у модифікації збережених процедур, функцій та тригерів направлених на оптимізацію та покращення якості їх виконання. При виконанні даних операцій діють правила рефакторингу програмного коду, але необхідно враховувати усі наслідки таких змін. Найпростішими прикладами є зміна назв збережених процедури з метою спрощення розуміння її призначення, а більш складним прикладом може бути заміна алгоритму обробки інформації.

Остання група операцій перетворення, що не входять до операцій рефакторингу, направлена на змін її семантики, адже вони можуть бути виправдані тільки в тому разі, якщо узгоджуються із замовником програмного забезпечення.

Одним з підходів уникнення проблем рефакторингу є використання тестів. Фактично доцільно виділити напрямки написання тестів:

- перевірка схеми бази даних;
- перевірка способів використання структур бази даних у програмних додатках;
- контроль введених даних;
- перевірка процедур та функцій, що зберігаються, на повну реалізацію покладеної на них бізнес логіки.

Для багатьох фахівців підхід, який передбачає тестування бази даних, є незвичним, тому їм доводиться стикатися з певними проблемами під час проведення рефакторингу бази даних. Однією з проблем є не відповідність персоналу з методикою тестування баз даних. Другою основною проблемою є недостатній асортимент інструментальних засобів для тестування бази даних. Такий інструментарій надається розробниками СУБД, але не завжди є можливість його придбати. Але є програмні засоби з відкритим кодом (Open Source Software - OSS), а саме:

- DBUnit (dbunit.sourceforge.net), що дозволяє управляти тестовими даними;
- SQLUnit, призначеного для тестування збережених процедур.

При виконанні рефакторингу бази даних не можливо обійтись без використання контролю версій. При кожній ітерації змін необхідне проведення контролю версій, адже це забезпечить відновлення втрачених файлів, повернення бази до робочої версії, надає можливість порівняння версій. Контроль версій існує двох типів – централізована та розподілена.

Традиційні системи контролю версій використовують централізовану модель, де всі функції контролю версій виконуються на загальному сервері. Якщо два розробники спробують змінити один і той же файл одночасно, без будь-якого методу управління доступом розробники можуть переписати роботу один одного. Централізовані системи контролю версій вирішують цю проблему за допомогою одного з двох варіантів «моделей управління версіями»: блокування файлів або злиття версій.

Розподілені системи контролю версій (DRCS) використовують одноранговий підхід, а не клієнт-серверний підхід централізованих систем. Вони, в загальному випадку, не потребують централізованого сховища: вся історія зміни документів зберігається на кожному комп'ютері, в локальному сховищі, і при необхідності окремі фрагменти історії локального сховища

синхронізуються з аналогічним сховищем на іншому комп'ютері. У деяких таких системах локальне сховище розташовується безпосередньо в каталогах робочої копії. Розподілений контроль версій здійснює синхронізацію шляхом обміну патчами (змінami) від тимчасових вузлів.

Ітерація рефакторингу повинна закінчувати контролем версій, що забезпечить відновлення втрачених файлів, повернення бази до робочої версії, надає можливість порівняння версій.

Використання рефакторингу забезпечує «чистоту», тобто ненадлишковість, не тільки коду, а й структури, архітектури, даних та потоків даних програмного проекту. Для проведення успішного рефакторингу баз даних необхідно дотримуватися чотирьох, правил, а саме:

- створювати повну копію бази даних та всього проекту;
- використовувати засоби автоматизованого контролю версій;
- будь-які зміни вносити у початковий код по створенню БД;
- використовувати засоби автоматизованого тестування.

Тобто рефакторинг тісно зв'язаний методами еволюційної методології, такими як – тестування та версіонування.

Висновки. Розглянуті підходи для рефакторингу баз даних застосовуються при використанні реляційних та об'єктно-орієнтованих систем управління баз даних. Їх обрання залежить від функцій програмного забезпечення та пріоритетів, що висуває команда розробників.

Для спрощення процесу рефакторингу на етапі обрання СУБД доцільно обирати об'єктно-орієнтовані СУБД, адже вони підтримують еволюційний підхід для створення програмного забезпечення, а от реляційні СУБД орієнтуються на послідовний підхід розробки.

1. Sadalage P. J., Ambler S. W. Refactoring databases: evolutionary database design [Text] (Addison-Wesley Signature Series) – Addison-Wesley Professional, 2006. – 384 p.
2. Fowler M., Sadalage P. J Evolutionary Database Design [Electronic resource] martinowler.com, May 2016: Proceedings. – Mode of access: <http://martinowler.com/articles/evodb.html> - Last access: 2017. – Title from the screen.
3. Fowler M. Refactoring [Electronic resource] Refactoring.com, May 2017: Proceedings. – Mode of access: <https://www.refactoring.com/> - Last access: 2017. – Title from the screen.
4. Robert C. Martin The Clean Coder: A Code of Conduct for Professional Programmers, 1st ed.; Prentice Hall, 2011. – 256p. – ISBN-13: 978-0137081073
5. Robert C. Martin Clean Architecture: A Craftsman's Guide to Software Structure and Design, 1st ed.; Prentice Hall, 2017. – 432p. – ISBN-13: 978-0134494166
6. Амблер С., Рефакторинг баз даних. Эволюционное проектирование / С. Амблер, П. Дж. Садаладж; пер. с англ. К. Птицын. – М.: Вильямс, 2016. – 368с. – ISBN-13: 978-5845911575
7. Лерми П. Рефакторинг SQL - приложений / П. Лерми, С. Фаро; пер. с англ. Ф. Гороховский – М.: Символ, 2009. – 336с. – ISBN-13: 978-5932861455
8. Макконнелл С. Совершенный код. Мастер-класс – М.: Русская Редакция, Microsoft Press, 2017. – 896 с. – ISBN-13: 978-5750200641
9. Gary McLean Hall Adaptive Code: Agile coding with design patterns and SOLID principles, 2nd ed.; Microsoft Press, 2017. – 448p. – ISBN-13: 978-1509302581
10. Месарош Д. Шаблоны тестирования xUnit. Рефакторинг кода тестов – М.: Вильямс, 2009. – 832с. – ISBN-13: 978-5845914484
11. Дэйт К. Дж. Введение в системы баз данных – М.: Вильямс, 2017. – 1328с. – ISBN-13: 978-5845907882
12. Тарасов С. СУБД для программиста. Базы данных изнутри – М.: Соломон, 2015. – 320с. – ISBN-13: 978-2746673830
13. Schwartz B., Zaitsev P., Tkachenko V. High Performance MySQL: Optimization, Backups, and Replication, 3rd ed.; O'Reilly Media, 2012. – 826p. – ISBN-13: 978-1449314286

УДК 004.42(07)

Христинець Н.А., Багнюк Н.В., Матейчук Н.В., Дзюбинський В.В.
Луцький національний технічний Університет

АНАЛІЗ PHP-ШАБЛОНІЗАТОРІВ ТА ВИКОРИСТАННЯ TWIG У DRUPAL 8

Христинець Н.А., Багнюк Н.В., Матейчук Н.В., Дзюбинський В.В. Аналіз PHP-шаблонізаторів та використання Twig у Drupal 8. У статті розкрито суть шаблонізаторів для відокремлення логіки від представлення, розглянуто їх основні переваги та недоліки. Здійснено порівняння характеристик поширених шаблонізаторів: Blade, Mustache, Smarty, Twig і Volt. Обґрунтовано використання Twig в якості двигуна шаблонів у Drupal 8.

Ключові слова: шаблонізатор, PHP, HTML, Twig, Drupal 8, шаблон, фреймворк, Blade, Mustache, Smarty, Volt.

Христинець Н.А., Багнюк Н.В., Матейчук Н.В., Дзюбинський В.В. Анализ PHP-шаблонизаторов и использования Twig в Drupal 8. В статье раскрыта суть шаблонизаторов для отделения логики от представления, рассмотрены их основные преимущества и недостатки. Проведено сравнение характеристик распространенных шаблонизаторов: Blade, Mustache, Smarty, Twig и Volt. Обосновано использование Twig в качестве двигателя шаблонов в Drupal 8.

Ключевые слова: шаблонизатор, PHP, HTML, Twig, Drupal 8, шаблон, фреймворк, Blade, Mustache, Smarty, Volt.

Khrystynets N.A., Bahniuk N.V., Mateychuk N.V., Dzyubinsky V.V. Analysis of PHP Templates and Using Twig in Drupal 8. The article discusses the essence of templates for separating logic from presentation, examining their main advantages and disadvantages. A comparison of the characteristics of common templates: Blade, Mustache, Smarty, Twig and Volt. The use of Twig as a template engine in Drupal 8 has been substantiated.

Keywords: template, PHP, HTML, Twig, Drupal 8, Template, Framework, Blade, Mustache, Smarty, Volt.

Постановка проблеми. Вже багато років іде дискусія на рахунок шаблонізаторів. Одні заявляють, що це хороша практика в програмуванні, а інші – що це не має виражених переваг, а тільки ускладнює розробку web-додатків. В роботі розглянуто розгорнуті характеристики популярних шаблонізаторів, переваги їх використання для різноманітних задач.

Аналіз досліджень. Антон Шевчук в своїй статті «Шаблонизаторы и PHP» [3] розкритикував існуючий популярний шаблонізатор – Smarty. Він заявив, що Smarty не вирішує головної задачі шаблонізатора – відділення логіки від представлення, адже Smarty дає можливість використовувати php код в шаблонах. Ще одним недоліком шаблонізатора він виділив складний синтаксис, який дуже часто є незрозумілим для верстальщиків і дизайнерів. Аналізуючи цю проблему, нами було виділено групу шаблонізаторів, що мають зручний синтаксиси та застосування.

Виклад основного матеріалу й обґрунтування отриманих результатів. Однією із основних характеристик шаблонізатора є можливість опису класу програм, що займаються заміною строкових послідовностей деякими даними. Слово «шаблонізатор» часто сприймається синонімом слова «подання». Заочно вважається, що якщо у проектах застосовується шаблонізатор, то у них відділено логіку представлення від бізнес-логіки. Існує думка, що шаблонізатори створені для дизайнерів/верстальників, а програмісту вони не потрібні. Насправді ж, використання шаблонізаторів полягає в тому, щоб зробити поділ логік зручніше, що надає деякі розширені функції готового веб-продукту. Можна використовувати будь-який шаблонізатор або можна взагалі обійтися без нього, але при цьому варто добре відділяти представлення від логіки. І навпаки, можна використовувати Smarty, але при цьому остаточно все заплутати. Тому, використання конкретного шаблонізатора єх позитивним рішенням для створення веб-продукту.

Поділ HTML і PHP

PHP і HTML тісно взаємодіють: PHP може генерувати HTML, а HTML може передавати інформацію PHP. Проте, складні чергування PHP і HTML ставали занадто складними для сприйняття. Зміна дизайну взагалі перетворювалася на кропітку роботу. Код дуже часто повторювався на різних сторінках і знайти його відповідний кусок було складно. Ще складніше було витягнути його звідти і вставити в новий проект.

Врешті програмісти прийшли до розуміння, що потрібно код HTML відокремити від PHP коду. Саме тоді з'явилося поняття шаблонізатора. Фактично, це є програмне забезпечення, що дозволяє використовувати html-шаблони для генерації кінцевих html-сторінок. Основна мета використання шаблонізатора – це відділення представлення даних від виконуваного коду. Часто це необхідно для забезпечення можливості паралельної роботи програміста і дизайнера-

верстальника. Використання шаблонізатора покращує читаність коду і внесення змін у зовнішній вигляд, коли над проектом працює група розробників.

PHP шаблонізатор корисний для проектів, що включають роботу розробників та дизайнерів.

Для складних програм необхідно розділити всі шари, починаючи з бізнес-логіки та закінчуючи презентацією. Це була основна причина створення шаблону дизайну MVC, який пізніше був прийнятий для веб-додатків. Проекти, створені в останні роки, просунуті на багатьох рівнях і вимагають сильної взаємодії з розробниками та дизайнерами. Обидва повинні співпрацювати, тому дуже важливим є пошук спільної мови для власної компетенції.

Переваги використання шаблонного двигуна

1. Відокремлення

Коли код додатка зростає і стає нечитабельним, розробники продумують структуру файлів проекту. Відокремлення моделі, перегляду та контролера вимагається за схемою дизайну, однак розробникам та дизайнерам все-таки потрібно працювати над їх загальним кодом. І шаблонний файл є найкращим місцем для такої співпраці.

2. Читабельність

Логіка та презентація, об'єднані в файли проекту, часто не є достатньо чіткими. Це виглядає непривабливо, і для дизайнера це дуже важко зрозуміти. Розробники розділяють код на окремі файли, оскільки обидва шари використовують різні технології, наприклад PHP та HTML. Найчастіше бекенд і фронтенд коди повинні підтримувати різні фахівці. Бекенд розробник PHP не зацікавлений в коді CSS, який використовується для стилізації веб-сторінки і, з точки зору дизайнера, те саме стосується написання запиту SQL на зворотному боці. Ось чому шаблони краще для всіх.

3. Кешування

Хороші двигуни шаблонів пропонують кешування для статичних файлів. Спеціальний синтаксис надає безліч функцій, але його потрібно проаналізувати та скласти з вихідною мовою. Шаблонні движки, такі як Smarty та Twig, мають кращу продуктивність зі скомпільованими файлами, що є ще одним кешуванням для веб-додатків.

4. Співпраця

Розробникам і дизайнерам доведеться працювати разом. Для складних проектів потрібні багато людей і досвідчені фахівці. Розробникам і дизайнерам потрібні свої ділянки в проекті, щоб працювати краще. Використання популярних інструментів допомагає їм усім зрозуміти один одного та легко співпрацювати.

5. Фільтри, модифікатори та багато інших

Шаблонні двигуни мають вбудовані функції, які дозволяють легко обробляти популярні сценарії. Якщо постає необхідність скоротити текст, форматувати рядок або дату, встановити значення за замовчуванням та багато інших, то рішення пропонують модифікатори та фільтри для простих завдань, які також дозволяють створювати власні плагіни та розширення. Всі ці приклади – це ще один шар абстракції, який допомагає проекту.

6. Легко вивчити

Розробники PHP дуже часто кажуть, що "PHP є мовою шаблону, тому немає сенсу вивчати іншу". З огляду на всі переваги, безумовно, добре вивчити мову двигуна шаблону, особливо, якщо мови набагато простіше для тих розробників, які раніше вивчали PHP.

Дизайнери можуть навчитися простому та інтуїтивно зрозумілому синтаксису шаблонів набагато легше порівняно з цілою складною мовою. Друк змінних, умовних інструкцій або циклів є дуже природним і простим у використанні навіть для нетехнічної людини.

7. Менший шанс зламати більше

Чим більше людей працюють з тим самим кодом, тим більша ймовірність того, що щось не так. Щоб запобігти конфліктам, розробники створили версію коду. Наступним кроком, що обмежує можливість фатальної помилки, є доступ до зони обмеженого доступу. Дизайнерам не потрібно читати та редагувати бекенд код. Місце для їх роботи обмежується файлами шаблонів.

Чому варто використовувати шаблонізатори

Усі зазначені аргументи є аналізом використаних літературних джерел, вони є результатом багаторічного досвіду роботипрофесіоналів. Можна не погодитися з ними. Але, з іншого боку, важко погодитися з використанням PHP як шаблону в кожній ситуації.

Порівняння шаблонізаторів

До шаблонізаторів застосовувалися такі критерії: вони повинні бути написані на PHP, активно підтримуватися і бути визнаними спільнотою PHP. При виборі шаблонізатора потрібно враховувати наступні фактори: синтаксис, логіка, розширюваність, документація, активність розробки, підтримка спільноти і продуктивність.

Blade

Цей шаблонізатор використовується в Laravel - PHP-фреймворку, що почав своє життя в 2011 р і став одним з популярних PHP-фреймворків. Причиною швидкості Blade є невеликий список регулярних виразів для заміни.

Mustache

Mustache доступний для практично необмеженої кількості мов, в тому числі і PHP. Також він містить мінімум логіки: заміна, цикл foreach, перевірка на null.

Smarty

Smarty з'явився в початку нульових, до цих пір розвивається і конкурує з більш молодими проектами.

Twig

Даний шаблонізатор знайшов свою популярність завдяки Фаб'єну Потенцеруб, який впровадив його в систему представлень фреймворка Symfony. Проте, Twig може бути впроваджений практично в будь-який проект, тобто незалежно від фреймворка.

Volt

Volt використовується в фреймворку Phalcon (фреймворк, написаний на C і поширюваний як PHP-розширення). З недоліків можна відзначити лише те, що Volt можна використовувати тільки в Phalcon, тобто немає можливості використовувати в проекті на іншому фреймворку.

Отже, найбільш оптимальним варіантом шаблонізатора є Twig. Він не залежить від фреймворка, запускається дуже швидко, містить досить функціоналу, має відмінну документацію і активно розвивається. Як відомо, у Drupal 8 шаблонізатор Twig використовується замість PHPTemplate. Тепер замість звичних нам *.tpl.php файлів шаблонів використовуються *.html.twig файли.

TwigvsPHPTemplate

Розглянемо основні особливості Twig та його відмінності від PHPTemplate.

Виведення змінної:

PHPTemplate:<?phpprint\$variable; ?>

Twig:{{ variable }}

Оператор If:

PHPTemplate:

```
<?php
if($variable_1=="1"): ?> <div>
<?phpprint$variable_2; ?>
</div>
<?phpendif; ?>
```

```
<?phpif(!empty($variable)):
$variable2= 'value';
endif; ?>
```

```
<?phpif(isset($variable)):
$variable2= 'value';
endif; ?>
```

```
<?phpif($variable< 0):
$variable2= 'value';
endif; ?>
```

Twig:

```
{% ifvariable_1 %}
<div>
{{ variable_2 }}
</div>
{% endif%}
```

```
{% ifvariableisnotempty%}
set variable2 = 'value '
{% endif%}
```

```
{% ifvariableisdefined %}
set variable2 = 'value '
{% endif%}
```

```
{% ifvariable< 0 %}
set variable2 = 'value'
{% endif%}
```

Надання змінній значення:

PHPTemplate:
<?php \$variable= 'some_value'; ?>

Twig:
{% setvariable = 'some_value' %}

Створення масиву:

PHPTemplate:
<?php \$my_array= array(1,2,3,4); ?>
<?php \$my_array= array('!element1'=>\$var1,
'!element2'=>\$var2); ?>

Twig:
{% setmy_array = [1,2,3,4,%]
{% setmy_array = {'!element1': var1,
'!element2': val2} %}

Цикли:

PHPTemplate:
<?phpforeach(\$usersas\$user) {
} ?>

Twig:
{% foruserinusers %} {%
endfor%}

ДебагінгTwig-шаблонів

Для того, щоб увімкнути можливість дебажититwig шаблони, необхідно у файлі *services.yml* у блоці налаштувань *twig.config* параметру *debug* надати значення *true*. Для зручності розробки можна також параметру *cache* встановити значення *false*. В цьому разі не потрібно буде після змін у шаблонах робити очищення кешу.

```
twig.config:  
  debug: true  
  cache: false
```

Встановлення власного шаблону для елемента

Розглянемо, як виглядає встановлення власного шаблону для елемента:

```
functionmy_theme_theme($existing, $type, $theme, $path) {  
  returnarray(  
    'block__system_menu_block'=>array(  
      'template'=> 'custom-menu-template',  
    ),  
  );  
}
```

de block__system_menu_block — елемент, для якого потрібно використати шаблон;
custom-menu-template — назва шаблону (ім'я файлу шаблону буде виглядати так: *custom-menu-template.html.twig*).

Переваги використання Twig у Drupal 8

Шаблонізатор Twig дозволяє веб-дизайнерам без будь-яких навичок в PHP модифікувати розмітку веб-сторінки. Це спрощує темізацію в Drupal 8. Це прекрасна альтернатива обробнику шаблонів PHPTemplate.

Стислість

Існують проекти, для яких реалізований програмний код на мові PHP. Але, в порівнянні з нею, Twig має більш стислий синтаксис, що робить шаблони більш читабельними. Він більш зрозумілий і простіший в компіляції. Крім того, Twig дає можливість створювати більш ефективні шаблони, завдяки синтаксису, який дуже схожий на синтаксис HTML.

Швидкість

Висока швидкість завантаження сайту дає змогу миттєво отримати необхідну інформацію. Швидкий процес розробки сайту також має вагомe значення для успіху всього проекту. Одною з цілей Twig є його швидкість. Для її досягнення Twig компілює шаблони в PHP-код. Крім того, вищезгаданий, оптимізований та зручний для читання синтаксис також допомагає веб-дизайнерам швидко виконувати свою роботу.

Безпека

Коли мова йдеться про безпеку, це має значення в будь-якому випадку і в будь-якому місці. Безпека веб-сайту настільки ж важлива, як і безпека процесу його розробки. У цьому аспекті Twig має кілька унікальних можливостей.

1. HTML escaping, або автоматичні виведення тексту у форматі HTML з міркувань безпеки можна застосувати у якості автоматичного виведення для частини коду або повністю для всього шаблону.

2. «Пісочниці, у середовищі яких робник визначає певний обмежений набір фільтрів, тегів та методів об'єктів. Користувач має доступ до нього, що дозволяє використовувати Twig як мову шаблонів для додатків, де користувачі можуть самостійно змінювати дизайн шаблону. Twig здатний оцінювати та автоматично ізолювати будь-який ненадійний код шаблону. Можна застосовувати «пісочниці» для окремих або для всіх шаблонів.

3. Бібліотека Twig повністю протестована та схвалена спільнотою. Вона є стабільною і нею можна користуватись при створенні різних проектів (додатків або сайтів).

Розширюваність

Ця корисна функція може задовольнити всі ваші потреби, чи то вони прості, чи складні. Завдяки гнучкості Twig і відкритій архітектурі можна додати нові функції, фільтри, теги, навіть синтаксис та ключові слова. Можливість розширення забезпечує розробників та веб-дизайнерів кращими інструментами для створення структури їх проекту в Drupal 8.

Наслідування

Найбільш потужною частиною Twig є наслідування шаблонів. Після впровадження Twig не доведеться копіювати файли батьківських шаблонів у власні користувацькі шаблони.

Висновки та перспективи подальшого дослідження.

Серед розглянутих шаблонізаторів в Drupal 8 найкращим варіантом є Twig. Він надає широкі можливості для розробки і до того ж він мультиплатформенний. Саме тому його вибрали розробники Drupal у своїй останній версії замість попереднього .tpl.php.

1. Addison Berry, Angela Byron, Bruno De Bondt. Using Drupal (2nd Edition). – O'Reilly. 2012. – 500p.
2. Al-Darwish, N.: PageGen: An Effective Scheme for Dynamic Generation of Web Pages. Information and Software Technology 45(10), 15 July 2003, Pages 651-662
3. Cynthia McCourt. Drupal: The Guide to Planning and Building Websites. – Wrox. 2011. – 504 p.
4. Douglas Vernon Denny. Drupal 7 Webform Cookbook. – Packt Publishing. 2012. – 276 p.
5. Fabien Potencier. Templating Engines in PHP (переклад), Templating engines in PHP — Follow-Up (переклад)
6. Jennifer Hodgdon. A Programmer's Guide to Drupal. – O'Reilly. 2012. – 114p.
7. Ric Shreves, Brice Dunwoodie. Drupal 7 Bible. – Wiley. 2011. – 768 p.
8. Smarty 3.1.29 Released — 2015.
9. Trevor James. Migrating to Drupal 7. – Packt Publishing. 2012. – 158p.
10. Байрон А., Берри Э. и др. Drupal: создание и управление сайтом / Пер. с англ. – СПб.: Символ-Плюс, 2010. – 576 с., ил.
11. Ромашов В.Р. CMS Drupal: система управления содержимым сайта. – СПб.: Питер, 2010. – 256 с., ил.
12. <http://anton.shevchuk.name/php/php-template-engine/>
13. <https://drudesk.com.ua/blog/funktsionalni-mozhlyvosti-twig>

УДК 004.65

Чорнобай К. Ю., Грибков С.В.

Національний університет харчових технологій

РОЗРОБКА БЮДЖЕТНОЇ СИСТЕМИ БІОМЕТРИЧНОЇ ІДЕНТИФІКАЦІЇ З ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ

Чорнобай К. Ю., Грибков С. В. Розробка бюджетної системи біометричної ідентифікації з використання нейронних мереж. В роботі проведено дослідження методів і підходів по створенню бюджетної системи біометричної ідентифікації. Система має архітектуру клієнт-сервер, клієнтська частина буде кроссплатформеною та орієнтована для мобільних пристроїв. Для обробки зображення використовується бібліотека з відкритим кодом Open Source Computer Vision Library та більшу частину по ідентифікації буде здійснюватися з використанням нейронних мереж.

Ключові слова – системи захисту, біометрична ідентифікація, обробка зображення, нейронні мережі.

Чорнобай К. Ю., Грибков С. В. Разработка бюджетной системы биометрической идентификации по использованию нейронных сетей. В работе проведено исследование методов и подходов по созданию бюджетной системы биометрической идентификации. Система имеет архитектуру клиент-сервер, клиентская часть будет кроссплатформенной и ориентирована для мобильных устройств. Для обработки изображения используется библиотека с открытым кодом Open Source Computer Vision Library и большую часть по идентификации будет осуществляться с использованием нейронных сетей.

Ключевые слова - системы защиты, биометрическая идентификация, обработка изображения, нейронные сети.

Chornobay K. U., Hrybkov S.V. Development of budget biometric identification system using neural networks. The paper focuses on the methods and approaches to create the budgetary system of biometric identification. The system have client-server architecture, the client part will be cross-platform and oriented for mobile devices. To process the image, an Open Source Computer Vision Library will be used and most of the identification will be done using neural networks.

Keywords - protection systems, biometric identification, image processing, neural networks.

Актуальність. Проблема захисту інформаційних ресурсів стала найактуальнішою в сучасному кібернетичному суспільстві, що вже не може уявити своє життя без використання мобільних пристроїв та доступу до ресурсів мережі Інтернет. Постійне вдосконалення пристроїв зв'язку та засобів спілкування дозволяє людині підвищити свою комунікацію. За рахунок багатьох ресурсів мережі Інтернет люди зберігають доволі багато корпоративної та приватної інформації на різних серверах, що мають різні ступені захисту. Проблеми захисту інформації на сьогодні одна з актуальних проблем, адже постійно зростають обсяги втраченої комерційної та конфіденційної інформації за рахунок несанкціонованого доступу. Основним фактором втрати інформації залишається людський фактор, бо користувач, який має доступ до інформації, може втратити засоби ідентифікації особи або їх можуть викрасти чи підробити.

Актуальним напрямом дослідження та розробка біометричних засобів аутентифікації, що базуються на використанні різних технічних та програмних засобів.

Аналіз останніх досліджень і публікацій. Проблемам захисту інформаційних ресурсів, а також його забезпечення присвячено багато наукових й практичних робіт вітчизняних та зарубіжних спеціалістів з даного напрямку. Авторами роботи [1] розглянуто основи інформаційної безпеки в комп'ютерних мережах та інформаційно-обчислювальних системах. У роботі [2] висвітлюють основні особливості захисту інформаційних ресурсів, що функціонують у корпоративних мережах, а також запропоновано підходи оцінки ефективності їх захисту. Роботи [3, 4] присвячені проблемам захисту конфіденційної та комерційної інформації у бездротових локальних мереж, наведено типи основних загроз та атак, а також описано комплекс заходів щодо їх усунення та підвищення якості безпеки. Автори роботи [5] дослідили способи забезпечення і реалізації аутентифікації та авторизації у web-орієнтованих системах. Автор роботи [6] розглядає різні стандарти та протоколи аутентифікації, підходи їх використання у web-додатках. У роботі [7] розглядаються проблеми реалізації механізмів аутентифікації та авторизації у web-орієнтованих системах, а також запропоновано підхід створення модуля аутентифікації та авторизації користувачів на основі JWT маркерів. Робота [8,9] присвячена огляду основних видів біометричної ідентифікації та їх реалізації, а також розглянуто усі їх недоліки та перспективи розвитку. У роботі [10] автори виділяють актуальний напрямок біометричної ідентифікації за обличчям людини, а також розглянуто методи ідентифікації за геометрією обличчя, адже риси обличчя і форма черепа кожної людини індивідуальні.

Виділення невирішених раніше частин загальної проблеми. Більшість розробок, що мають впровадження та ефективно використовуються, як правило направлені на ідентифікацію за

відбитками пальців та райдужної оболонки ока, що використовуються у різних системах захисту. Такі системи захисту, як правило використовують для доступу до пристроїв або у певне приміщення. Організація доступу з використання біометричної ідентифікації для доступу до інформаційних систем мають обмежене застосування та, як правило, у спеціалізованих організаціях. Сьогодні стає можливим використання в якості засобів зчитування біометричних параметрів для ідентифікації мобільні засоби комунікацій з фотоелементами. Крім цього, певні фірми виробники мобільних пристроїв обладнують свої пристрої засобами ідентифікації за відбитками пальців, голосом, обличчям. Деякі фірми запатентували системи біометричної ідентифікації та їх використовують в своїх брендах, а саме Sony, Apple. Але актуальною задачею залишається створення системи біометричної ідентифікації на основі обличчя людини, геометрії обличчя, рис обличчя і форм черепа, що є індивідуальними для кожної людини. Впровадження біометричних засобів ідентифікації потребують багато вкладень у апаратно-технічні засоби, тому доступні не всім державним та приватним установам.

Формлювання мети дослідження. Необхідно провести дослідження та аналіз підходів по створенню бюджетної системи біометричної ідентифікації для підвищення ефективності захисту доступу до інформаційних ресурсів різного рівня. Інтеграція системи повинна проходити без особливих проблем, а також система повинна складатися з двох основних елементів, а саме: серверної частини, що буде забезпечувати саме ідентифікацію та надання доступу до інформаційних ресурсів; клієнтської частини, що буде функціонувати на мобільному пристрої та забезпечувати знімання та передачі зчитаних біометричних параметрів.

Виклад основного матеріалу досліджень. Управління роботою організації багато в чому зводиться до запобігання причин, що перешкоджають виконанню її функцій. Одна з таких причин - порушення повноважень доступу на території об'єктів або до джерел конфіденційної інформації і обчислювальних ресурсів організації. Щоб розмежувати доступ співробітників і клієнтів організації на її об'єкти або до інформації, що становить певну значимість, існують спеціальні автоматизовані системи.

Процес ідентифікації полягає в пред'явленні користувачем ідентифікатора та зіставлення його зі зразками ідентифікаторів всіх користувачів в системі. Найбільшого поширення ідентифікація отримала в областях, де безпосередній контакт людини з системою не передбачений, наприклад, в системах розпізнавання осіб. Як перевірка особистості користувача при доступі до ресурсів сучасних комп'ютерних систем не використовується з причин значно меншій надійності в порівнянні з аутентифікацією.

Існуючі системи аутентифікації користувачів:

- паролі системи (найпростіший і найпоширеніший спосіб);
- системи РКІ (криптографічні сертифікати);
- системи одноразових паролів;
- біометричні системи.

В основі більшості механізмів аутентифікації користувачів лежать ключі та паролі, тому даний спосіб має найбільшого поширення. Через свою «відкритість», а також велику кількість ПО для злому паролів, дана система є найбільш вразливою.

Для того, щоб система функціонувала з необхідними рівнями швидкодії і надійності, необхідно правильно обрати, що використовувати в якості ідентифікатора і аутентифікатора.

На основі проведених досліджень створено бюджетну систему біометричної ідентифікації. В основі системи покладено клієнт-серверну архітектуру. Серверна частина підтримує роботу бази даних зображень облич, а також сукупність модулів для здійснення ідентифікації та ведення статистики ідентифікації. Клієнтська частина буде реалізована в якості кроссплатформенного додатку, що буде забезпечувати зчитування зображення обличчя, формування відправлення фото до серверної частини для ідентифікації та отримання відповіді про її результат.

Планується впровадження клієнтського додатку на мобільні пристрої під управлінням операційної системи сімейства Android та обладнаних фотокамерою, а також на комп'ютери різної архітектури з підключеною веб-камерою під управлінням операційних сімейств Windows або Linux.

Клієнтська частина є кроссплатформенною та орієнтована в першу чергу на використання мобільних пристроїв з камерами. Для обробки зображення буде використовуватися бібліотека з відкритим кодом Open Source Computer Vision Library (OpenCV), що створена була саме для роботи з пристроями відео зйомки та складної обробки зображень, а також має модулі по

ідентифікації об'єктів. Ця бібліотека була створена компанією Intel®, щоб затвердити загальний стандартний інтерфейс комп'ютерного зору для програмних продуктів в цій області та для сприяння їх більшого розвитку, а також для створення нових моделей використання персональних комп'ютерів. Модулі бібліотеки OpenCV забезпечують [12]:

- функції введення/виведення при роботі як із стандартними пристроями, так і цифровими пристроями фото та відео зйомки;
- виконання алгоритмів фільтрації та перетворення обробки зображення;
- повний цикл використання методів інтелектуального аналізу для обробки зображення;
- визначення та опис плоских примітивів;
- аналіз руху об'єкту та його відстеження;
- реалізацію більшості алгоритмів пошуку об'єктів на зображенні;
- управління пристроями зчитування відеоінформації;
- обробку трьохвимірних об'єктів.

Основним алгоритмом ідентифікації людини на основі зображення її обличчя буде послідовність наступних кроків:

- виявлення факту присутності людини на зображенні;
- виділення контуру фігури людини;
- визначення ракурсу голови (анфас, профіль);
- виділення з зображення обличчя основних параметрів за шаблонами;
- порівняння параметрів з еталонами та ідентифікація.

Для реалізації основних кроків алгоритму достатньо функціоналу бібліотеки OpenCV, але, на думку авторів, для ідентифікації користувачів за зображенням обличчя запропоновано використовувати метод нейронних мереж, що дозволяє вирішувати складні завдання по ідентифікації візуальних та аудіо образів. Нейронні мережі не програмуються в звичному сенсі цього слова, вони навчаються. Можливість навчання - одне з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. В процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. Це означає, що в разі успішного навчання мережа зможе повернути вірний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних або пошкоджених частково даних. В якості образів можуть виступати різні за своєю природою об'єкти: символи тексту, зображення, зразки звуків тощо. При навчанні мережі пропонуються різні зразки образів із зазначенням того, до якого класу вони відносяться. Зразок, як правило, представляється як вектор значень ознак. При цьому сукупність усіх ознак повинна однозначно визначати клас, до якого належить зразок. У разі, якщо ознак недостатньо, мережа може співвідносити один і той же зразок з декількома класами, що невірно. Після закінчення навчання мережі їй можна пред'являти невідомі раніше образи і отримувати відповідь про належність до певного класу.

Коли мережі пред'являється якийсь образ, на одному з її виходів повинен з'явитися ознака того, що образ належить цьому класу. У той же час на інших виходах повинен бути ознака того, що образ даного класу не належить. Якщо на двох або більше виходах є ознака приналежності до класу, вважається, що мережа «не впевнена» в своїй відповіді. Нейронні мережі базуються на паралельній обробці інформації і мають здатність до самонавчання, тобто отримувати обґрунтований результат на підставі даних, що не зустрічалися в процесі навчання.

До властивостей та переваг нейронних мереж доцільно віднести:

- масовий паралелізм;
- розподілене представлення інформації і обчислення;
- можливість самонавчання та схильність до виділення загальних характеристик;
- можливість пристосовуватися до різних вхідних даних та задач;
- при наявності певних відхилень чи помилок дає ефективний результат;
- мінімальні вимоги.

Основними параметрами при аналізі зображення обличчя людини будуть наступні:

- основні контури обличчя, очей, брів, рота, носа, що є характерними і подібними для усіх людей, але не однаковими, та слугують для побудови певної карти контурів для первинної ідентифікації та характеризуються яскравими переходами на зображенні;

- яскравість, що дає змогу виявити основні контури, а також виявити особливості кожної людини при однакових умовах освітлення;
- колір, що слугує точною ознакою об'єкта за рахунок того, що несе додаткову інформацію про відтінок шкіри.

Перевагою створеної системи ідентифікації є досить гнучка інтеграція з будь-якою інформаційною системою чи комплексом, що мають базу даних під управлінням основних СУБД. Система блокує доступ та не дає пройти навіть до аутентифікації та ідентифікації у системах в які вона інтегрується. Також до переваг необхідно віднести незначні вимоги до її налаштування. Мобільна частина написана під систему Android, та забезпечує функції фотографування та передачі інформації через мережу до серверної частини. Такий спосіб підвищує ідентифікацію та аутентифікації в інформаційних системах, а також є дуже бюджетним. До недоліків необхідно відвести те, що при здійсненні фото все залежить від якості знімку, що не завжди можуть забезпечити бюджетні мобільні телефони нижньої цінової категорії, але якщо здійснювати знімки саме з них, а потім проходити ідентифікацію з такого мобільного пристрою то забезпечується певний додатковий ефект захисту. Також необхідно відмітити про недоліки розпізнавання на основі нейронної мережі, адже, як показали випробовування, завжди існує невірний варіант ідентифікації, але для уникнення такої ситуації доцільно робити тренування мережі на основі декількох знімків обличчя та бажано при різному освітленні. Але все ж таки підвищити якість ідентифікації можливо лише використовуючи змішані підходи біометричної з іншими видами ідентифікації.

Для забезпечення високого рівня захисту необхідно використовувати комбіновані біометричні системи аутентифікації, що передбачає поєднання деяких типів біометричних характеристик для ідентифікації. Це дозволяє задовольнити найсуворіші вимоги до ефективності системи аутентифікації. Наприклад, аутентифікація за відбитками пальців може легко поєднуватися зі скануванням руки. Така структура може використовувати всі види біометричних даних людини і може застосовуватися там, де доводиться форсувати обмеження однієї біометричної характеристики. Комбіновані системи є більш надійними з точки зору можливості імітації біометричних даних людини, так як важче підробити цілий ряд характеристик, ніж фальсифікувати одну біометричну ознаку.

Висновки. Використання створюваної системи біометричної ідентифікації дуже широка та орієнтована на державні та комерційні об'єкти з обмеженим фінансуванням, такі як дитячі садочки, школи, університети, лікарні та ін. Одним з напрямків використання даної системи є використання її для аутентифікації в сучасних web-додатках та для доступу до комерційних бездротових локальних мереж Wi-Fi, що стане додатковим захистом від несанкціонованого доступу.

1. Емельянова Н.З., Партыка Т.Л., Попов И.И. Защита информации в персональном компьютере. - М.: Форум, 2009. - 368 с.
2. Кононова В.О., Грибков С.В., Харкянен О.В. Оцінка засобів захисту інформаційних ресурсів / В.О. Кононова, С.В. Грибков, О.В. Харкянен // Вісник Національного університету «Львівська політехніка», Львів : Видавництво Львівської політехніки, №806, 2014, с. 99 – 105.
3. Визавитин О. И. Практика защиты информации в Wi-Fi сетях на основе современных программно-аппаратных средств // Молодой ученый. — 2016. — №5. — С. 182-184.
4. Чорнобай К.Ю., Грибков С.В. Аналіз та шляхи вирішення проблем захисту комерційних бездротових локальних мереж Wi-Fi / К.Ю. Чорнобай, С.В. Грибков // Вісник Національного університету «Львівська політехніка», серія «Автоматика, вимірювання та керування», Львів : Видавництво Львівської політехніки, №880, 2017, с. 99 – 103.
5. Шелупанова А.А., Груздева С.Л., Нахаева Ю.С. Аутентификация. Теория и практика обеспечения доступа к информационным ресурсам. / А.А. Шелупанова, С.Л. Груздева, Ю.С. Нахаева – М. : Горячая линия – Телеком, 2009. – 552 с.
6. Обзор способов и протоколов аутентификации в веб-приложениях / Д. Выростков – Режим доступа до ресурсу : <https://habrahabr.ru/company/dataart/blog/262817/>.
7. Олійник Г.В., Грибков С.В. Використання JWT маркерів для аутентифікації та авторизації користувачів у WEB – додатках / Г.В. Олійник, С.В. Грибков // Вісник Національного університету «Львівська політехніка», серія «Автоматика, вимірювання та керування», Львів : Видавництво Львівської політехніки, №880, 2017, с. 86 – 93.
8. Anil K. Jain, Kathik Nandakumar, Biometric Authentication: System Security and User Privacy. IEEE Computer, November 2012, IEEE Computer Society. All rights reserved. Reprinted with permission.
9. Брагина Е.К., Соколов С. С. Современные методы биометрической аутентификации: обзор, анализ и определение перспектив развития / Е.К. Брагина, С. С. Соколов // Вестник астраханского государственного технического университета. – 2016. – №1(61). – С. 40–44.
10. Вакуленко А., Юхин А. Биометрические методы идентификации личности: обыкновенный выбор // Сборник научных трудов 1 Международной научно-практической конференции "Мировой опыт применения

- биометрических решений в составе комплексных систем безопасности". – К. : "Информация-Украина". – 2006. – С. 79–82.
11. Фисенко В. Т., Фисенко Т. Ю., Компьютерная обработка и распознавание изображений: учеб. пособие. – СПб : СПбГУ ИТМО, 2008. – 192 с.
 12. Bradsky G., Kaehler A. Learning OpenCV – O'Reilly, 2008. – 508 p.

УДК 004.78

Яцюк С. М., Яцюк А. В., Муляр В. П.

Східноєвропейський національний університет імені Лесі Українки

ВИКОРИСТАННЯ ФРЕЙМВОРКІВ ДЛЯ РОЗРОБКИ WEB-ДОДАТКІВ НА МОВІ PHP

Яцюк С. М., Яцюк А. В., Муляр В. П. Використання фреймворків для розробки Web-додатків на мові PHP.

В статті проведено аналіз сучасних фреймворків для розробки Web-додатків на мові PHP, розроблено критерії вибору PHP-фреймворків. Зроблено висновки щодо доцільності застосування PHP-фреймворків.

Ключові слова: Прийняття рішень, PHP-фреймворк, MVC, фреймворк, Front-End, Back-End, CSS, MVC.

Яцюк С. М., Яцюк А. В., Муляр В. П. Использование фреймворков для разработки Web-приложений на языке PHP. В статье проведен анализ современных фреймворков для разработки Web-приложений на языке PHP, разработаны критерии выбора PHP-фреймворков. Сделаны выводы о целесообразности применения PHP-фреймворков.

Ключевые слова: Принятие решений, PHP-фреймворк, MVC, фреймворк, Front-End, Back-End, CSS, MVC.

Yatsiuk Svitlana, Yatsiuk Andrew, Mulyar Vadim. The use of frameworks for the development of web-applications in the language of PHP. In the article an analysis of modern frameworks for the development of Web-applications in the language of PHP, developed criteria for the choice of PHP-frameworks. Conclusions regarding the feasibility of using PHP-frameworks are made.

Keywords: Decision Making, PHP Framework, MVC, Framework, Front-End, Back-End, CSS, MVC.

Постановка проблеми. Розвиток Інтернету нерозривно пов'язано з проектуванням сайтів. Масова поява сайтів спровокувала проблему їх якості. Популярність створення Web-ресурсів сприяла розробці різних систем і програм, які спрощують процес написання сайту. Також вони допомагають підвищити ефективність роботи, а також дозволяють розробнику сфокусуватися на основній логіці програми. Такі технології, як PHP, Java, Microsoft.Net, MySQL, Oracle, Microsoft SQL Server і розроблені на їх основі фреймворки – це каркаси системи або підсистеми, що можуть включати допоміжні програми, мови сценаріїв – і все, що полегшує розробку й об'єднання різних компонентів. Фреймворки за останній час набрали популярність, і стали базовою платформою для розробки Web-додатків.

Аналіз літературних джерел. Проаналізувавши інформацію з мережі Інтернет, можна побачити що існує сотні фреймворків для створення Web-додатків. Тому досить складно зробити вибір фреймворку, так як кожен з них має велику кількість привабливих функцій та доповнень. А неправильний вибір фреймворку може стати основною причиною невдачі проекту. Тема вибору фреймворку погано розкрита в мережі Інтернет та друкованих джерелах. Тому доцільно виділити критерії, на які потрібно спиратися при виборі фреймворку, щоб не допустити непоправної помилки. Також потрібно оцінити які фреймворки є найбільш простими та дозволяють створити веб-сайт швидше за все, а які надають сайту найбільше можливостей й гнучкості.

Мета даної роботи полягає у дослідженні можливостей найбільш поширених сучасних PHP фреймворків, порівнянні їх та пошуку найкращого та найзручнішого з них, розробці критеріїв вибору фреймворків.

Виклад основного матеріалу дослідження. Фреймворки – це програмні продукти, які спрощують створення і підтримку технічно складних проектів. Фреймворк містить тільки базові програмні модулі, а всі специфічні для проекту компоненти реалізуються розробником на їх основі. Тим самим досягається не тільки висока швидкість розробки, але й продуктивність і надійність створеного проекту[2].

Фреймворк відрізняється від бібліотеки тим, що бібліотека може бути використана в програмному продукті просто як набір підсистем близької функціональності, не впливаючи на архітектуру основного програмного продукту і не накладаючи на неї ніяких обмежень. Фреймворк диктує правила побудови архітектури проекту.

Одним з головних переваг при використанні фреймворків є те, що веб-проекти часто використовують стандартизовану структуру організації компонентів. Створення структури при розробці на фреймворках значно спрощується. По суті, фреймворк – це множина конкретних і абстрактних класів, а також визначення способів їх взаємовідношень. Конкретні класи зазвичай реалізують взаємні відношення між класами, а абстрактні класи являють собою точки розширення, в яких каркаси можуть бути використані або адаптовані. [1,2,5]

Вибір правильного фреймворка для проекту на PHP – перший крок для його успішної реалізації.

PHP – найбільш популярна серверна скриптова мова програмування. З тих пір як вона з'явилась, а сталося це в 1995 році, складність веб-проектів зросла настільки, що вже немає сенсу писати код для всього з нуля. Щоб якось структурувати процес розробки, були створені фреймворки.

Проаналізуємо і порівняємо 10 найпопулярніших PHP-фреймворків[3,4,6]. Згідно аналізу розробимо критерії вибору відповідного фреймворку для свого проекту.

FUELPHP

FuelPHP - повноцінний PHP-фреймворк, який підтримує не тільки архітектуру MVC, але ще і HMVC (Hierarchical - ієрархічна MVC). У нього є опціональний клас під назвою Presenter (колишня назва - ViewModel) між шарами Controller і View, в якому міститься логіка, необхідна для генерації уявлень.

Фреймворк FuelPHP є модульним, його можна розширювати. У фреймворку приділено увагу питанням безпеки: реалізована фільтрація даних, що вносяться користувачем, запитів, а також даних, що виводяться. У нього є свій фреймворк авторизації і безліч інших просунутих функцій, а також досить велика документація.

SLIM

Slim - мікрофреймворк, що ідеально підходить для невеликих проектів або програм, де повноцінний фреймворк здасться зайвим. Його використовують PHP-розробники для створення RESTful API і сервісів. Серед функцій Slim - кешування HTTP на стороні клієнта, URL-маршрутизація, шифрування сесій і cookie, а також миттєві повідомлення по HTTP-запитах. Документація повна і зроблена якісно.

PHALCON

Phalcon був створений в 2012 році і швидко став популярним серед PHP-розробників. Його вважають дуже швидким, так як він написаний на C і C ++, щоб досягати найвищого можливого рівня оптимізації продуктивності. Але не хвилюйтеся, вчити C не доведеться - вся функціональність укладена в PHP-класи, які можна використовувати для будь-яких цілей.

Так як Phalcon спочатку був створений як розширення на C, його архітектура оптимізована на низькому рівні, що значно знижує витрату ресурсів, типово для додатків, заснованих на схемі MVC. Phalcon не тільки підвищує швидкість виконання, але і знижує рівень витрат ресурсів. У цього PHP-фреймворку є і багато інших чудових функцій: універсальний автозавантажувач, менеджмент ресурсів, безпека, переклад, кешування і т. д. Документація для Phalcon досить розширена, а використовувати його нескладно.

Цікавий факт: ще в 2014 році Phalcon був другим за популярністю фреймворком, згідно з даними sitepoint.com. А вже в 2015-му він помітно здав позиції і перемістився ближче до кінця списку.

CAKEPHP

Фреймворку CakePHP вже десять років, а він все ще популярний. Роботі з ним легко навчитися, а шаблонування - швидке і налаштовується. Вбудована функція CRUD дуже допомагає при взаємодії з базою даних. В останньому релізі - CakePHP 3.x - покращився управління сесіями і модульність (вони роз'єднали кілька компонентів), а також розширилися можливості створення більшої кількості окремих бібліотек.

Серед проектів, реалізованих на CakePHP, - сайти BMW і Hyundai. Вибирайте його, якщо вашому веб-додатку потрібно високий рівень безпеки, тому що функцій для цього тут більш ніж достатньо:

- валідація введення;
- захист від атак з використанням впроваджуваного SQL (SQL injection);
- запобігання міжсайтового скриптинга;
- захист від підробки міжсайтових запитів і багато іншого.

SYMFONY2

Це PHP-фреймворк з відкритим кодом, який використовується для розробки web-додатків, іншими словами – це інструменти і методології, які спрощують процес створення web-проектів. Характерні особливості, що відрізняють даний фреймворк, полягають в його швидкості, гнучкості, масштабованості і стабільності. Він цілком може бути використаний в якості великого

повномасштабного web-додатку або ж для невеликої функціональної потреби вже наявного проекту. Symfony2 підтримує безліч PDO-сумісних СУБД.

ZEND FRAMEWORK 2

Zend Framework 2 - фреймворк з відкритим вихідним кодом, що використовується для розробки веб-додатків і сервісів на PHP 5.3 +. Він використовує на 100% об'єктно-орієнтована код і більшість нових функцій PHP 5.3: простору імен, пізніше статичну зв'язування, лямбда-функції і замикання. Zend - надійне рішення з безліччю варіантів конфігурації. Зазвичай його не рекомендують використовувати для невеликих додатків, а ось для великих проектів це саме те.

Серед функцій Zend Framework 2: інструменти криптографічного кодування, простий у використанні drag-and-drop-редактор з підтримкою фронтенд-технологій (HTML, CSS, JavaScript), миттєва онлайн-налагодження, інструменти для unit тестування, майстер конфігурації бази даних. Творці цього фреймворка врахували методологію Agile, що дозволяє створювати високоякісні програми для корпоративних клієнтів.

У партнерах Zend - IBM, Microsoft, Google і Adobe. Пік тому Zend оголосив про наступне великому релізі - Zend Framework 3, оптимізованому під PHP 7, але все ще підтримує PHP 5.5 і вище. Всі чекали новинку ще восени 2015 го, а й зараз, в квітні 2016- го третьої версії досі немає.

Yii 2

Вибирайте Yii, щоб підвищити продуктивність сайту. Він швидше за всіх інших PHP-фреймворків, так як використовує технологію завантаження на вимогу (lazy loading). Yii 2 повністю об'єктно-орієнтована і заснований на принципі Don't-Repeat-Yourself («не повторювати»), так що основа для коду буде чиста і логічна.

Yii 2 інтегрований з jQuery і поставляється з набором AJAX функцій. Механізми скіннінга і вибору тим тут прості, так що фреймворк сподобається тим, хто раніше займався фронтенд-розробкою. Тут також є потужний генератор вихідного коду - Gii, який сприяє об'єктно-орієнтованого програмування і швидкому прототіпуванню, а також надає веб-інтерфейс, в якому можна інтерактивно генерувати потрібний код [7].

ASP.NET

Платформа ASP.NET MVC базується на взаємодії трьох компонентів: контролера, моделі та подання. Контролер приймає запити, обробляє користувацький вхід, взаємодіє з моделлю і повертає користувачеві результат обробки запиту.

Модель представляє сукупність слоїв, що описує логіку організації даних в додатку. Подання отримує дані з контролера і генерує елементи призначеного для користувача інтерфейсу для відображення інформації.

ROR

Це повноцінний, багаторівневий фреймворк для побудови Web-проектів, що використовують бази даних, заснований на архітектурі Модель-Представлення-Контролер (Model-View-Controller, MVC).

Динамічний AJAX-інтерфейс, обробка запитів і видача даних в контролерах, предметна область, відображена в базі даних, – для всього цього Rails надає однорідне середовище розробки на Ruby. Все, що необхідно для початку роботи – база даних і веб-сервер.

STRUTS

Це середовище Open Source для розробки Web-проектів за допомогою технологій Java Servlet and Java Server Pages (JSP). Struts сприяє розробці додатків з архітектурою, що базується на платформі Model-View-Controller, відомої як Model 2. Спостерігає складові частини фреймворка і їх функції визначені в Struts:

1. Сервлет-контролер (controller servlet) координує запити до відповідних дієвих класів (action), створених розробниками програми.
2. Бібліотеки JSP тегів підтримуються сервлетом-контролером, що допомагає розробникам у створенні інтерактивних Web-додатків, заснованих на шаблонах.
3. Допоміжні класи для підтримки розбору (parsing) XML, автоматичного керування властивостями JavaBeans класу-дії, і інтернаціоналізації повідомлень і запитів.

Таблиця 1. Критерії вибору фреймворків

Фреймворк							
Критерій	Zend	CakePHP	Symfony2	Yii	ASP.NET	ROR	Struts
Мова	PHP	PHP	PHP	PHP	ASP.NET	Ruby	Java
Шаблон проектування MVC	+	+	+	+	+	+	+
Застосування ORM	+	+	+	+	OR-M незалежна	+	+
Інтеграція з БД	+	+	+	+	***	+	***
Механізми інтернаціоналізації і локалізації	+	+	+	+	***	+	+
Використання шаблонів при створенні інтерфейсів користувача	+	+	***	+	+	+	+
Створення і перевірка форм	+	+	+	+	+	+	+
Управління доступом на основі ролей	+	+	***	+	+	+	ні
Використання AJAX	+	+	***	+	+	+	+
ЧПУ (Friendly URL)	+	+	+	+	+	+	ні
Модульне тестування (unit-testing)	+	+	***	+	+	+	+
Система кешування	+	+	+	+	+	+	ні

Висновки. Одна з головних переваг фреймворка – зручна розробка нестандартних проектів. Жоден великий нестандартний проект (наприклад, twitter.com, фотобанк з онлайн купівлею фотографій, сайт знайомств і т.д.) не розроблений на готовій CMS – вони для цього не призначені. Усі оригінальні проекти розробляють на фреймворках. Web-проект, розроблений за допомогою фреймворку, розвивається динамічно. При зміні вимог змінюється і сайт, достатньо лише замінити окремий блок (модуль), створити новий розділ або внести новизну в дизайн.

В перспективі плануємо удосконалювати та вивчати нові фреймворки, які створюються розробниками на ринку і використовувати їх для вдалого створення Web-проектів.

1. Результаты тестирования шести ведущих фреймворков на производительность [Електронний ресурс] – Режим досту- пу до ресурсу: <http://www.alrond.com/ru/2007/jan/25/rezultaty-testirovaniya-6-frameworks/>.
2. Використання PHP фреймворків в розробці сайту [Електронний ресурс] – Режим доступу до ресурсу: <http://ukrbukva.net/page,5,39718-Ispol-zovanie-PHP-freymvorkov-v-razrabotke-saiyta.html>.
3. Сравнение каркасов веб-приложений [Електронний ресурс] – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Сравнение_каркасов_веб-приложений.
4. Обзоры Web-фреймворков [Електронний ресурс] – Режим доступу до ресурсу: <https://praktikatech.wordpress.com/category/обзоры-web-фреймворков/>
5. Что такое фреймворк? [Електронний ресурс] – Режим доступу до ресурсу: <http://www.dbhelp.ru/what-is-framework/page/>.
6. Десять причин избегать тяжеловесных фреймворков, а также лишних зависимостей в проекте [Електронний ресурс] – Режим доступу до ресурсу: <http://eax.me/avoid-frameworks/>.
7. Yii framework [Електронний ресурс] – <http://workmake.ru/veb-razrabotka/freymvorki/obzor-freymvorka-yiinachinaem-razbiratsya-cto-i-kak-rabotaet>

УДК 004.05(075.8)

І.Є. Андрушак, Ю.Я. Матвіїв, В.А. Кошелюк, О.М. Сіваковська¹⁾, В.П. Марценюк²⁾

Луцький національний технічний університет¹⁾

Університет Бельсько-Бяли, Польща²⁾

DEFINITION OF INTERACTION AT THE SAME TIME ATTENDED TRACKS ON THE MODULE OF THE VECTOR OF THE TRANSITION.

І.Є. Андрушак, Ю.Я. Матвіїв, В.А. Кошелюк, О.М. Сіваковська, В.П. Марценюк. Визначення взаємовпливу одночасно утворених тріщин на модуль вектора переміщень. У статті розглянуто особливості розв'язання актуальної науково-технічної задачі, а саме – розробки методів оцінки докритичного розтріскування скловолоконних композитних матеріалів за параметрами сигналів акустичної емісії та показані основні переваги і недоліки їх впровадження.

Ключові слова: скловолоконні композити; акустична емісія; докритичне розтріскування; дефект; силовий критерій.

И.Е. Андрушак, Ю.Я. Матвеев, В. А. Кошелюк, А.Н. Сиваковська, В.П. Марценюк. Определение взаимовлияния одновременно образованных трещин на модуль вектора перемещений. В статье рассмотрены особенности решения актуальной научно-технической задачи, а именно - разработки методов оценки критическому растрескиванию стекловолоконных композитных материалов по параметрам сигналов акустической эмиссии и показаны основные преимущества и недостатки их применения.

Ключевые слова: стекловолоконные композиты; акустическая эмиссия; критическому розтріскування; дефект; силовой условий.

I.Ye.Andrushchak, Yu.Ya. Matthew, V.A. Koshelyuk, O.M. Sivakovskaya, V.P. Martsenyuk. **Definition of interaction at the same time attended tracks on the module of the vector of the transition.** The article considers the peculiarities of the solution of the actual scientific and technical problem, namely the development of methods for evaluating subcritical cracking of fiberglass composite materials according to the parameters of acoustic emission signals, and shows the main advantages and disadvantages of their implementation.

Key words: fiberglass composites; acoustic emission; subcritical stripping; defect; power criterion.

Formulation of the problem. The need for mankind in new constructional materials grows so fast that the pace of studying the properties of new materials practically does not have time to create them. This is especially true for composite materials (CM), which due to the appearance of new high-strength fibers with low specific gravity and high specific strength, which is significantly higher than that of aluminum alloys and steels, corrosion, chemical, heat, fire and moisture resistance; good thermal and electrical properties are widely used in all spheres of human activity. The development of world-wide production of CM has led to the fact that there is currently no area of technology or the fields of science and management, wherever they would use these materials. CM is particularly widely used in shipbuilding, machine building, aerospace engineering, chemical industry, residential and industrial construction, military industry.

Setting up tasks. In the world practice in the fields of fiber-optic composite materials, there are currently no generally accepted methods for determining the characteristics of their static crack resistance. This is due to the problems of determining the stages of macro-destruction of such composites, which may be due to various factors: the filtration of fibers from the matrix, the destruction of the matrix and the fibers itself, the extraction of the latter, and others like that. To study these processes and in order to effectively detect them, lately, the phenomenon of acoustic emission is increasingly being used. For this purpose, a number of parameters of the acoustic emission signals are used to obtain the necessary information and interpretation of the results of scientific research.

As the world practice shows, the definition of the stages of cracking of composites, and especially the dynamics of its development, is at the stage of formation. That is why the research carried out in this paper is aimed at solving a scientific and technical problem relevant for a number of industries in Ukraine - the development of effective applied methods of quantitative evaluation of the parameters of the pre-crystalline cracking of fiberglass composite materials, which are most prevalent in various spheres of industrialization in Ukraine.

The ability to choose the orientation of fibers and combine their various types in conjunction with the selection of matrix material allows you to create a CM with all the best strength and elastic properties. It should be noted that the most widespread among them is glass fiber, since their cost is the lowest. This led to the greatest use of glass fiber CM in the above technical areas. However, regarding the strength characteristics of fiberglass composites, it should be emphasized that they are essentially different for

different types of reinforcing component and matrix. Therefore, the relevance of the problem of studying the strength properties and crack resistance of such CM can not be overestimated today, as they have not yet been studied in detail.

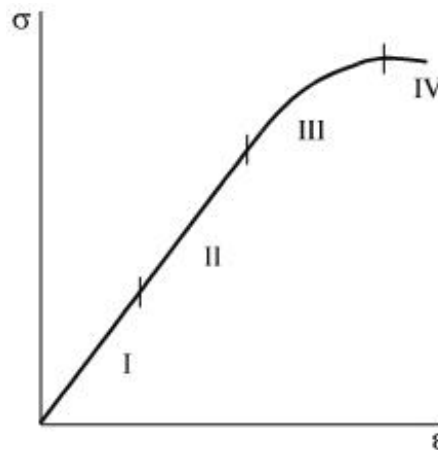
Analysis of recent researches and publications. Often during the operation of products or individual elements of constructions there are situations in which the formation of microcracks occurs in the localized volume of structural material. Such processes are discrete with little time duration and intervals between cracks formation events. For their diagnosis, for example, the AE method, it is necessary to be able to distinguish between sequential events, since this depends on the assessment of the destruction of the CM. In this aspect, the essential role played by the location of cracks in relation to their characteristic dimensions. Of course, non-destructive control and technical diagnostics of CM on the basis of AE phenomena require the establishment of appropriate correlation relations between the AE signals and the processes of formation and growth of various defects that arise under the influence of static and statically-variable charges in the material. They lead to the destruction of the matrix, fibers, filler, as well as the boundaries of the section between them. Therefore, it is especially important to insert the resolution of new techniques, taking into account the threshold of sensitivity of hardware, the specifics of the structure and mechanical characteristics of the CM.

In this article, an analysis of displacements is presented on the basis of a known solution of the dynamic problem of the theory of cracks on the formation of a internal insulated crack of normal separation and a static approach to the interaction of two arbitrarily oriented cracks. The results of the research are published in the works [1-3].

Basic material presentation. The analysis of the deformability of fibrous composites in the longitudinal direction uses mainly the classical theory of mixtures, direct numerical methods for analyzing the stress-strain state near the inclusions of a cylindrical shape, and so on.

The theoretical basis for the deformation of such composites in the cross-section is based on solutions of the flat theory of elasticity and plasticity for plates with single or interacting massive (most often circular and elliptical forms) inclusions in isotropic or anisotropic media for an ideal or partial contact between them. Handling of regularly reinforced fibrous composites according to this approach best describes two-period solutions [4].

In the case of applying forces in parallel to the one-leg direction of stacking fibers in a metal matrix in papers [5, 6], there are four stages of deformation of the CM (dr. 1.1).



Dr. 1. Schematic curve of deformation of composite material with a metal matrix:

- I - elastic fiber, elastic matrix; II - elastic fiber, plastic matrix;
- III - plastic fibers and matrix; IV - destroyed fiber.

At the first stage there is elastic deformation at the same time both the matrix and the fibers. The experimentally obtained modulus of elasticity in this case is very consistent with the prediction of the theory of mixtures:

$$E_C = E_R f_R + E_R f_R, \quad (1)$$

where E_R , E_M і E_C – modulus of elasticity of the strengthening component (filler), matrix and composite material, respectively; f_R і f_M - bulk particles of fiber and matrix. If the porosity or the third phase is absent, then $f_F + f_M = 1$.

It is established in [7] that equation (1) corresponds theoretically to the lower boundary of the module, when the Poisson coefficients of two phases are very close to each other. The same equation holds for the case of elastic compression of a composite. Higher than calculated values of the elastic modulus can be obtained in the case of nonconformity of the Poisson coefficients. Thus, the simple additivity rule of the properties of mixtures, applied in the field of elastic deformation, proved to be very successful for predicting elastic deformation of the composite material. The second stage of deformation of a unidirectional fibrous metallic composite concerns an area in which the fiber deforms elastically, and the matrix is plastic. Since the volumetric fraction of the fiber is usually high and it has a modulus of elasticity much higher than the matrix, then the elastic modulus E_2 at this stage, almost coincides with the module E_1 the first stage of deformation (1). More precise dependence on the elastic modulus of the composite material along the reinforcement direction (E_{11}) can be obtained on the basis of the similar (1) assumption that this modulus of elasticity is equal to the sum of effective strain curves of the two phases, taking into account their volumetric particles:

$$E_{11} = E_F f_F + (d\sigma_M / d\varepsilon_M)_G f_M, \quad (2)$$

where $d\sigma_M / d\varepsilon_M$ – effective coefficient of deformation strengthening of the matrix.

This coefficient is usually much smaller than the fiber elastic modulus. Therefore, it can be neglected and in this connection, an effective modulus of elasticity on the second (elastic-plastic) part of the deformation curve in most cases can be represented by an expression:

$$E_{11} = E_F f_F. \quad (3)$$

It is necessary to pay attention to the fact that though E_{11} almost equal to the value E_M (85 % E_M for 50% volumetric boron content in the aluminum-boron composition), it is not a real (real) modulus of elasticity. The deformation of the composite material in the elastic-plastic region causes some steel elongation, and the relaxation of the applied stress causes the fiber to be elastically stretched. The matrix thus has residual compressive stresses. The cyclic deformation of the composite in this (second) region of deformation causes the effect of the hysteresis effect, which can significantly change the effect of the previous thermomechanical treatment of the material.

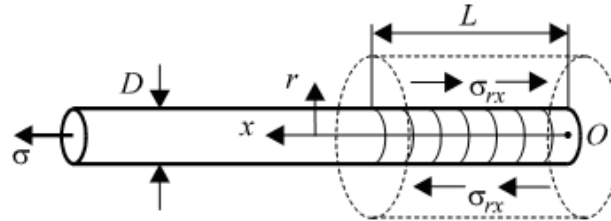
The third stage of deformation occurs when both components (fiber and matrix) undergo plastic deformation and are the usual flat stretch of two phases. In this case, the deformation process of the components in the composition material can fundamentally differ from the deformation of the separately loaded components, as far as neck formation or other type of non-homogeneous plasticity is concerned. For some plastic fibers, the beginning of neck formation is significantly delayed due to the sub-actuation of the matrix. For example, in the composite system, silver-steel fixed [8] increase the strength of the composition above the level of the combined strength of composite materials.

The fourth stage corresponds to a period of gradual destruction of high-strength fibers. At this stage, the matrix transfers the load from the ends of the destroyed fiber to the undiluted parts and there is a flowing flow around the open pores or cracks. Usually this stage ends with the global destruction of composite material. Due to the ductile plasticity of the matrix and the dynamic loads that cause the multiple division of the fiber into a part which does not yet lead to catastrophic destruction, the composite material may remain generally intact if the volume fraction of the fibers is less than a certain critical the size of This stage corresponds to the deformation stage of the composite material containing inside the torn fibers. In the presence of short fibers, the matrix transfers the load as a result of the appearance of tangent stresses on the surface of the fibers near the ends of the broken fiber. These shear stresses are caused by a gradient of normal stresses when moved to the fiber in the radial direction, since the fibers have a higher effective deformation slope than the matrix, which is also apparent from the balance of forces in the direction of the axis x (dr. 2):

$$dP / dx = 2\pi r \sigma_{rx}, \quad (4)$$

where P – the tension of the fiber in the direction x ; $\sigma_{rx} \sim \tau$ – displacement stress on the fiber-matrix separation surface; r – the radial distance from the axis of the fiber.

An approximate solution of the problem of transmission of tangents in an elastic state has the form:



Dr. 2. The forces affecting the reinforcing fiber: σ – tensile stress acting on a cross-section $\pi D^2 / 4$;

τ – tensile shear stress acting on the surface area $\pi D l$

$$P = \varepsilon_F A_F \left(1 - \frac{(\cosh \beta)(l/2 - 1)}{(\cosh \beta)l/2} \right) \quad (0 < x < l/2). \quad (5)$$

There P , l i ε_F – load, length and strain of fiber tension, respectively; A_F – cross-sectional area of the fiber; β – the ratio of length to fiber diameter (the influence of the geometry of the ends of the fibers on the intensity of the transfer of load to the fibers was not taken into account).

When loading composite materials with a metal matrix, when the fiber is deformed elastically, and the matrix is plastic, then in this case the elastic deformation of the matrix is limited to the values of the normal σ_y or shifted τ_y matrix strain stresses. If the deformation strengthening effect is small, then one can predict that σ_{rx} does not depend on x , and therefore, after integrating equation (4), taking into account that the newly formed end of the fiber after its rupture becomes unloaded, will give an expression for the longitudinal force:

$$P = P(0) + 2\pi r_0 \int_0^x \sigma_{rx} dx = 2\pi r_0 x \sigma_{rx}, \quad (6)$$

and then normal tension will be:

$$\sigma_{xx} = P / (\pi r_0^2) = 2\sigma_{rx} x / r_0. \quad (7)$$

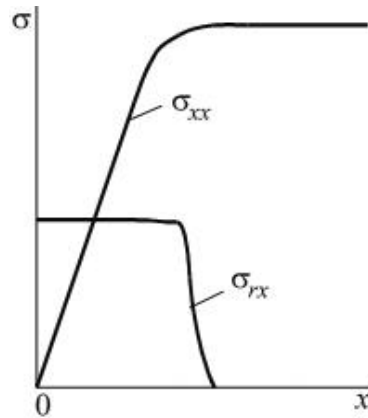
As can be seen from equation (7), the normal tensile stress increases linearly with the distance from the ends of the fiber, provided that the tensile shear stress is steady, as shown in Fig. 3. In the case of composite material with fibers of finite length (short), the minimum critical length of the reinforcing phase may be determined, in which the stresses may increase to the magnitude of the destructive stress. If ticked through σ_F the tensile strength of the fiber, then the critical length of the reinforcing phase L_c can be determined using the value of yield strength τ_y for the displacement of the matrix by tangential stresses. From equation (7), which describes the range of tangential stresses applied at each end of the fiber and which shows that in the middle of the finite fiber of length l tangential stresses σ_{rx} on its surface and normal tensile stresses associated with each other expression $\sigma_{xx} = \sigma_{rx} l / r_0$, we will determine for the boundary state ($\sigma_{xx} \sim \sigma_F$) an expression:

$$L_c = r_0 \sigma_F / \tau_y. \quad (8)$$

Just tension stretch σ_{xx} in the shattered fibers the magnitude corresponding to the stresses for the undiluted fibers will reach, the tensile stresses on the boundary of the separation in the matrix will again be equal to zero (dr. 3), as in the case of a composite material in which there is no damaged fibers.

The destruction of weak fibers in the composite leads to the following results:

1. The overall strength of the composite material in the cross-section in the place of the destruction of the fiber decreases;
2. Effective strength decreases due to the concentration of the static stress field near the crack caused by the broken fiber;
3. There is a certain shock load of the composite material and, accordingly, its retaining ability is reduced due to the generation of waves of dynamic stress during the discharge of the destroyed



Dr. 3. Distribution of normal σ_{xx} and tangent $\sigma_{rx} = \tau$ tension at the ends of the fiber

fiber.

The ability of the composition in the presence of shattered fibers is also related to the critical length of the load transfer on the fiber in the matrix according to equation (8).

In the zone of direct localization of the destruction of the fiber, the strength decreases by the value of the holding capacity of the destroyed element. At distances up to the critical loading load, the material's retaining capacity as a result of the destruction of the fiber decreases.

Conclusion. The processes of deformation of materials are accompanied by the formation of microproblems in zones where a sufficiently high concentration of disulocytic aggregates is achieved. The formation of such microdefects, accompanied by the emission of elastic waves. The registration of elastic waves AE makes it possible to study the kinetics of the processes of accumulation of defects, especially in areas of high concentration of stresses. To construct the calculation model of the emission of elastic waves during the formation of a crack in the zone of intense deformations for loading of the composite material, suppose that such a nubally-formed crack is of a form close to the circular one. To simplify the calculations, replace it with a disc-like split equilibrium crack. We also assume that due to the formation of this discoid crack, the breaking of the solidity of the material is accompanied by the unloading of free surfaces, formed from the initial level σ_0 (in a defect-free material) to zero, which occurs for some characteristic time τ_r . The unloading of the shores of the newly created crack causes fluctuations in its shores and, accordingly, the emission of elastic waves.

1. Skalsky V.R. Amplitude of Acoustic Emission Signals Caused by Cracking in a Composite Material / V.R. Skalsky, O.M. Sergienko, D.V. Rudavsky Yu.Ya. Matveyev // Machine Science. - 2007. - No.12. S.44-49.
2. Scalsky V. Diagnosis of the formation of dysplastic cracks in fibrous composites / V.Skalsky, O.Sergiyenko, Y.Matviev // Proceedings and Internews. conf. "Theory and practice of rational design, manufacture and operation of machine building structures", Lviv, October 22-24, 2008. - Lviv, KINPATRI LTD, 2008. - P. 73-75.
3. Skalsky V. Estimation of diagnostics of fibres failure in composite materials by the method of acoustic emission / V.Skalsky, O.Serhiyenko, D.Rudavskyy, Yu.Matvijiv // Proc. V Międzynarodowego Sympoz. Mechaniki Zniszczenia Materiaływ i Konstrukcji. Augustyw, 3 – 6 czerwca 2009. – P. 107 – 108.
4. Grigoluk E.I., Filhtinsky L. A. Perforated plates and sheaths. - Moscow: Nauka, 1970. - 556 pp.
5. Kelly A, Davies G.L. The principles of the reinforcement of metals // Metallurgical Reviews. – 1965. – 10, No.37. – P. 1 – 77.
6. Jech R.W., Mc Daniels D.L., Weeton J.W. Stress-strain behaviour of tungsten-fiber-reinforced copper composites. – NASA TN D-1881. – 1963. – 43 p.
7. Rosen B.W., Dow N.F., Hashin Z. Mechanical properties of fibrous composites. – NASA TN CR-31. – 1964. – 153 p.
8. Piehler H.R. Plastic deformation and failure of silver-steel filamentary composites // Trans. AIME – 1965. – 233, No.4. – P. 12–16.

УДК 303.09

¹Бомба А.Я., д.техн. н., ²Федонюк А. А. к. фіз.-мат. н.,

¹Рівненський державний гуманітарний університет

²Східноєвропейський національний університет імені Лесі Українки

ПРО УЗАГАЛЬНЕННЯ ОДНІЄЇ МОДЕЛІ ІНФОРМАЦІЙНОЇ БОРОТЬБИ

Бомба А.Я., Федонюк А. А. Про узагальнення однієї моделі інформаційної боротьби. Аналізується математична модель інформаційного впливу на соціум двох взаємовиключаючих ідеологій з введенням поняття запізнення в часі стосовно початку дії інформаційного потоку.

Ключові слова: математична модель, інформаційний потік, запізнення, адепт, соціум.

Бомба А.Я., Федонюк А. А. Об обобщении одной модели информационной борьбы. Анализируется математическая модель информационного влияния на социум двух взаимоисключающих идеологий с учетом понятия опоздания во времени относительно начала действия информационного потока.

Ключевые слова: математическая модель, информационный поток, опоздание во времени, адепт, социум.

Bomba A.Y., Fedoniuk A. A. On the generalization of one model of information struggle. The mathematical model of information influence on the society of two mutually exclusive ideologies with the consideration of the concept of delay in time regarding the beginning of the information flow.

Keywords: mathematical model, information flow, delay in time.

Постановка наукової проблеми.

Сучасні підходи до вивчення проблеми інформаційної безпеки з точки зору «людського фактору» можуть і повинні бути збагачені точними методами. До них відносяться в першу чергу використання відповідних математичних моделей, які будуються з використанням наукових досягнень в сфері соціології та психології. Літературний огляд та аналіз основних правових документів держави Україна вказує на те, що, на жаль, проблеми інформаційної безпеки зводяться в основному лише до питань її захисту від несанкціонованого доступу, або до питань силового запобігання розповсюдженню небажаної інформації [1].

Існуючі технології поширення інформації (або брехливої інформації) дають найширші можливості для діяльності недружніх держав, екстремістських спільнот, терористичних організацій і т.д. для пропаганди своїх інтересів і реалізації своїх цілей в усіх сферах життя людини і суспільства. У відповідності до цього однією з важливих задач забезпечення ІБ, крім визначення джерел, характеру та механізмів виникнення і поширення соціально значущих інформаційних загроз, є пошук методів парирования ІБ або їх ослаблення на основі більш тонких і науково обґрунтованих підходів. Як підкреслювалося в [8], серйозні системи моніторингу, аналізу та прогнозу ІБ з метою запобігання породжених ними негативних наслідків не повинні зводитися до обслуговування чисто технократичних підходів і методів до проблем інформаційної безпеки. Необхідно, щоб вони містили підсистеми моделювання і прогнозування цих процесів, включаючи ієрархію відповідних математичних моделей, забезпечених адекватним соціологічним «оснащенням».

Розвиток суспільства – процес безперервний і надзвичайно багатогранний. Створити ідеальну модель для прогнозування поведінки соціуму в тому чи іншому напрямку практично неможливо, але, якщо розглядати лише певні, конкретно вибрані аспекти життєдіяльності, то в такому разі створення математичних моделей призводить до абсолютно реальних поведінкових шаблонів. Створення та аналіз математичних моделей дуже корисна для суспільства річ, адже завдяки цьому можна з достатньою достовірністю прогнозувати соціальні процеси, аналізуючи реальні головні фактори впливу, та, зрештою, і керувати цими процесами шляхом правильно побудованої інформаційної політики держави, регіону, конфесії, організації тощо.

Аналіз досліджень.

У вибраній нами для аналізу та вдосконалення роботі [5], яку ми вважаємо базовою в силу найбільш ґрунтовного опису найпростішої математичної макромоделі розповсюдження інформаційної загрози [3,8] побудовані і досліджені моделі, що відповідають актуальній ситуації – ситуації інформаційного протистояння.

Пропоновані моделі отримані в розумних і інтерпретованих припущеннях. В рамках їх застосовності вони дозволяють вивчити характер розглядуваних процесів. Це дає можливість ставити і вирішувати завдання про знаходження оптимальних, в певному сенсі, способів їх організації [6].

Показано, що досліджувані процеси в силу їх нелінійності допускають не очевидні режими розвитку. Тим самим, навіть в найпростіших випадках з аналізу математичних моделей поширення інформаційної загрози і інформаційного суперництва визначаються змістовні характеристики, управління якими може стимулювати перебіг цих процесів в потрібному для їх учасників напрямку.

Модель одночасного розповсюдження двох взаємовиключаючих видів інформації (інформаційне протистояння)

Задачу окреслюють таким чином: нехай на певну сукупність людей чисельністю N_0 впливають два абсолютно виключаючі один одного інформаційні потоки Π_1 та Π_2 . Концепції Π_1 та Π_2 є взаємовиключаючими із-за специфіки свідомості людини, яка полягає в тому, що одна людина може мати лише одне переконання, яке стосується деякого конкретного питання, або ж просто ще не мати чітко сформульованого переконання щодо цього питання взагалі.

Нехай в деякий момент часу $t_0=0$ два джерела одночасно починають трансляти свою інформацію, в результаті чого обидва потоки поширюються в суспільстві.

Оскільки Π_1 та Π_2 не ідентичні, то цей процес розглядається як інформаційне протистояння (конкуренція). Загальну математичну модель такого процесу вибудовували багато авторів, зокрема [1,2,5].

В [5] описується динаміка цього інформаційного протистояння, тобто залежність від часу t величин $N_1(t)$ та $N_2(t)$ числа «адептів» які сприйняли інформацію, що розповсюджуються джерелами «1» або «2» відповідно. Там же теоретично визначають переможця в конкретному протистоянні. (Переможцем вважається та ідеологія, яка в момент повного охоплення аудиторії в деякий момент часу зуміла розповсюдити свою інформацію серед більшого ніш суперник числа членів суспільства, тобто величини $N_0/2$ – половини від загальної кількості досліджуваного соціуму N_0).

Основні модельні припущення робляться по аналогії з моделлю для одного інформаційного потоку [5]:

1. Кожен з потоків Π_1 та Π_2 розповсюджується серед суспільства по двох інформаційних каналах:
 - a) «зовнішній» по відношенню до соціуму. Інтенсивність розповсюдження інформації цим каналом для Π_1 характеризується параметром $\alpha_1 > 0$, а для Π_2 – $\alpha_2 > 0$, які рахуються такими, що не залежать від часу;
 - b) «внутрішній» – міжособистісне спілкування (його інтенсивність для Π_1 характеризується параметром $\beta_1 > 0$, а для Π_2 – $\beta_2 > 0$, які рахуються такими, що не залежать від часу). Впродовж такого спілкування вже завербовані ідеєю «1» адепти (їх число рівне величині $N_1(t)$, впливаючи на ще не завербованих членів (їх число рівне величині $N_0 - N_1(t) - N_2(t)$), вносять свій «особистий» вклад в процес вербування. Так само завербовані ідеєю «2» адепти (їх число рівне величині $N_2(t)$), впливаючи на ще не завербованих членів (їх число теж рівне величині $N_0 - N_1(t) - N_2(t)$), вносять свій «особистий» вклад в процес вербування.
2. Швидкість зміни числа адептів $N_1(t) - N_2(t)$ (тобто число завербованих в одиницю часу Π_1 та Π_2) складається з:
 - a. швидкостей зовнішнього вербування (вони пропорційні добутку інтенсивностей α_1 та α_2 на числа діючих членів ($N_0 - N_1(t) - N_2(t)$), тобто величинам $\alpha_1(N_0 - N_1(t) - N_2(t))$ та $\alpha_2(N_0 - N_1(t) - N_2(t))$, відповідно для Π_1 та Π_2 ;
 - b. швидкостей внутрішнього вербування (вони пропорційні добутку інтенсивностей β_1 та β_2 на числа діючих адептів $N_1(t)$ та $N_2(t)$ на числа не завербованих ($N_0 - N_1(t) - N_2(t)$), тобто є величинами $\beta_1 N_1(t)(N_0 - N_1(t) - N_2(t))$, та $\beta_2 N_2(t)(N_0 - N_1(t) - N_2(t))$, відповідно для Π_1 та Π_2 .

Число ще не завербованих членів рівна величині N_0 без врахування числа індивідуумів сприйнявших вже обидва види інформації, тобто $N_1(t) + N_2(t)$. Параметри α_1 , α_2 , β_1 та β_2 характеризують не лише інтенсивність інформаційного впливу, але ще і схильність до його сприйняття. Таким чином, ще не завербована до моменту часу t частина суспільства (її гіпотетичний «середній» представник спочатку нейтральний до обох потоків інформації Π_1 та Π_2) здобуває інформацію тим швидше, чим більші величини α_1 та β_1 , α_2 та β_2 . При цьому, навіть коли дія Π_1 заздалегідь сильніша дії Π_2 ($\alpha_1 > \alpha_2$, $\beta_1 > \beta_2$), частина членів суспільства все рівно сприймає Π_2 (немає повної монополії одного виду інформації по відношенню до іншої).

Підсумовуючи припущення 1 та 2, отримується модель (1):

$$\begin{aligned} \frac{dN_1}{dt} &= (\alpha_1 + \beta_1 N_1(t))(N_0 - N_1(t) - N_2(t)), & N_1(t_0 = 0) &= N_1(0) > 0, \\ \frac{dN_2}{dt} &= (\alpha_2 + \beta_2 N_2(t))(N_0 - N_1(t) - N_2(t)), & N_2(t_0 = 0) &= N_2(0) > 0 \end{aligned} \quad (1)$$

Система нелінійних звичайних диференціальних рівнянь є автономною динамічною системою другого порядку і може виступати базовою моделлю для досліджуваного процесу. З цієї системи можна при різних відомих параметрах N_0 , α_1 , α_2 , β_1 , β_2 та початкових значеннях $N_1(0)$ та $N_2(0)$ знайти аналітично і чисельно всі шукані характеристики. Цю систему автори [5] називають моделлю інформаційного протистояння.

Поділивши друге рівняння на перше отримується:

$$\frac{dN_2}{dN_1} = \frac{\alpha_2 + \beta_2 N_2}{\alpha_1 + \beta_1 N_1}$$

Звідси можна знайти загальне рішення системи у вигляді інтеграла:

$$\beta_2 N_2(t) = C(\alpha_1 + \beta_1 N_1(t))^{\beta_2/\beta_1} - \alpha_2$$

$$\text{де } C = [\alpha_2 + \beta_2 N_2(0)]/[\alpha_1 + \beta_1 N_1(0)]^{\beta_2/\beta_1}.$$

Мета та задача дослідження:

Використовуючи роботу [5] як базову, ми бачимо, що вона ґрунтовно описує модель протистояння двох інформаційних потоків в суспільстві, причому потоків, які є взаємовиключаючими і до того ж антагоністичними. В роботі подається динаміка процесу протистояння інформаційних потоків та аналізується момент перемоги однієї ідеології над іншою.

Але, тут же ми бачимо, що в роботі [5] використовуються модельні припущення, які не зовсім точно характеризують реальні суспільні процеси. Наприклад, для сучасного українського суспільства болючою є тема окремих районів Донецької та Луганської областей (ОРДЛО). Як відомо, вербування зі сторони т.зв. «руського мира» велося і в значній мірі ще до початку відомих подій 2014 року. Що ж стосується проукраїнської інформаційної роботи, то на той момент її практично зовсім не було.

В нашій роботі ми намагаємося побудувати математичну модель інформаційного протиборства для випадку коли виникає запізнення в часі для одного з інформаційних потоків. Тобто вносимо в систему диференціальних рівнянь, що є моделлю інформаційного протистояння, параметр τ , який і характеризує процес запізнення в часі.

Знаходимо розв'язки цієї системи для характерних точок і намагаємося проаналізувати поведінку соціуму при різному перебігу подій, а також спрогнозувати умови для перемоги інформаційного потоку при значному запізненні в часі.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Система диференціальних рівнянь із запізненням τ .

Використаємо модельні припущення, зроблені авторами роботи [5] (див. вище), але внесемо в ці припущення деякі зміни. Ми вважатимемо, що в момент часу $t=t_0$ функції N_1 та N_2 вже мають деяке конкретне значення (3).

$$\begin{cases} \frac{dN_1}{dt} = (\alpha_1 + \beta_1 N_1)(N_0 - N_1(t - \tau) - N_2(t - \tau)); \\ \frac{dN_2}{dt} = (\alpha_2 + \beta_2 N_2)(N_0 - N_1(t - \tau) - N_2(t - \tau)); \end{cases} \quad (2)$$

$$N_1(\tilde{t}) = N_{10}(\tilde{t}), \text{ де } -\tau \leq \tilde{t} \leq 0; \quad (3)$$

$$N_2(\tilde{t}) = N_{20}(\tilde{t}), \text{ де } -\tau \leq \tilde{t} \leq 0;$$

Розв'язки системи шукаємо у вигляді (4)

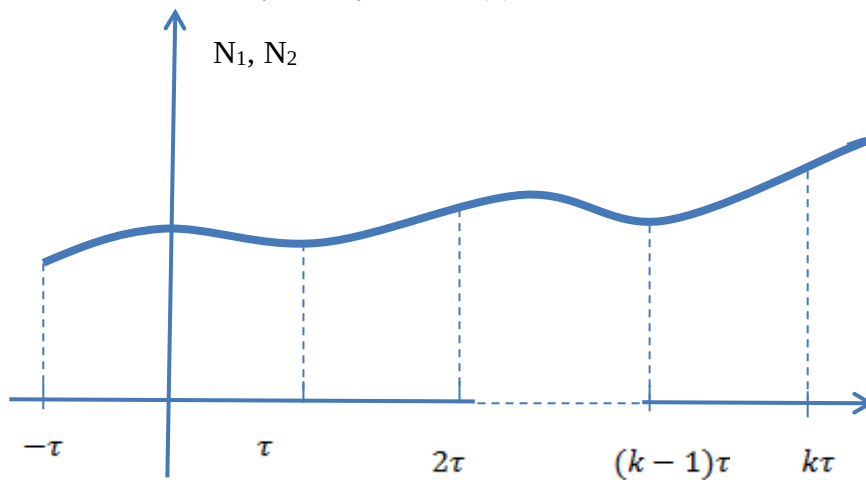


Рис.1. Залежність (гіпотетична) кількості індивідуумів N_1 або N_2 деякого соціуму від часу спостереження. Час розбитий на рівні інтервали τ для зручності аналізу.

$$N_i(\tilde{t}) = \begin{cases} N_{i1}(t), & 0 \leq t < \tau \\ N_{i2}(t), & \tau \leq t < 2\tau \\ \dots & \dots \\ N_{ik}(t), & (k-1)\tau \leq t < k\tau \end{cases} \quad (4)$$

де $i=1, 2$.

Розглянемо проміжок часу $0 \leq t < \tau$. Звідси слідує $N_i(t) = N_{i1}(t)$, $-\tau \leq t - \tau < 0$

Отже, $N_i(t - \tau) = N_i(\tilde{t}) = N_{i0}(\tilde{t}) = N_{i0}(t - \tau)$

Матимемо:

$$\begin{cases} \frac{dN_{11}}{dt} = (\alpha_1 + \beta_1 N_1) * f_1(t), \text{ де } f_1(t) = N_0 - N_{10}(t - \tau) - N_{20}(t - \tau); \\ \frac{dN_{21}}{dt} = (\alpha_2 + \beta_2 N_1) * f_1(t); \end{cases} \quad (5)$$

Розв'язуємо цю систему при умовах (6)

$$\begin{cases} N_{11}(0) = N_{10}, & N_{10} = N_{10}(0 - \tau) = N_{10}(-\tau) \\ N_{21}(0) = N_{20}, & N_{20} = N_{20}(0 - \tau) = N_{20}(-\tau) \end{cases} \quad (6)$$

Матимемо:

$$\frac{1}{\beta_1} \frac{d(\beta_1 N_1 + \alpha_1)}{dt} = f_1(t) dt;$$

$$\ln(\beta_1 N_1 + \alpha_1) = (\int_0^t f_1(\tilde{t}) d\tilde{t} + \tilde{C}_1) * \beta_1, \text{ позначимо } \beta_1 \tilde{C}_1 = \overline{C}_1$$

$$N_1(t) = C_1 e^{\int_0^t f_1(\tilde{t}) d\tilde{t}} - \frac{\alpha_1}{\beta_1}$$

$$\text{При } t=0 \quad N_{10} = C_1 e^0 - \frac{\alpha_1}{\beta_1}, \quad C_1 = N_{10} + \frac{\alpha_1}{\beta_1}$$

$$\begin{cases} N_{11}(t) = (N_{10} + \frac{\alpha_1}{\beta_1}) e^{\int_0^t f_1(\tau) d\tau} - \frac{\alpha_1}{\beta_1}, \\ N_{21}(t) = (N_{20} + \frac{\alpha_2}{\beta_2}) e^{\int_0^t f_1(\tau) d\tau} - \frac{\alpha_2}{\beta_2} \end{cases}$$

Тоді, для $\tau \leq t < 2\tau$ $N_i = N_{i2}$ $N_i(t - \tau) = N_{i1}(t - \tau)$

$$\begin{cases} \frac{dN_{12}}{dt} = (\alpha_1 + \beta_1 N_{12}) * f_2(t), \text{ де } f_2(t) = N_0 - N_{11}(t - \tau) - N_{21}(t - \tau); \\ \frac{dN_{22}}{dt} = (\alpha_2 + \beta_2 N_{22}) * f_2(t); \end{cases} \quad (5)$$

$$N_{12}(\tau) = N_{120}, \quad N_{120} = N_{11}(\tau)$$

$$N_{22}(\tau) = N_{220}, \quad N_{220} = N_{21}(\tau)$$

$$\begin{cases} N_{12}(t) = (N_{11} + \frac{\alpha_1}{\beta_1}) e^{\int_\tau^t f_2(\tau) d\tau} - \frac{\alpha_1}{\beta_1}, \\ N_{22}(t) = (N_{21} + \frac{\alpha_2}{\beta_2}) e^{\int_\tau^t f_2(\tau) d\tau} - \frac{\alpha_2}{\beta_2} \end{cases}$$

Аналогічно, на проміжку

$$(k-1)\tau \leq t < k\tau$$

$$N_i = N_{ik}(t) = (N_{ik} + \frac{\alpha_i}{\beta_i}) e^{\int_{(k-1)\tau}^t f_k(\tau) d\tau} - \frac{\alpha_i}{\beta_i}$$

В своїх попередніх дослідженнях [9,10] ми, використовуючи математичну модель одночасного впливу обох інформаційних потоків дослідили вплив православних церков двох конфесій (УПЦ КП та УПЦ МП) на український соціум в період незалежності. Використовуючи отримані в цих роботах коефіцієнти $\alpha_1=3*10^{-2}$, $\beta_1=4*10^{-6}$, $\alpha_2=3,5*10^{-3}$, $\beta_2=7,9*10^{-6}$ ми застосували розв'язки, отримані вище для реальної ситуації в суспільстві. Було виявлено співпадіння з реальністю із зовсім незначною кореляцією даних. Найкращий результат отриманий був при запізненні в часі $\tau=3-7$ років. Ми хочемо набрати більше статистичних даних і скорегувати час запізнення до більш конкретного терміну. Цим ми займемося в найближчій перспективі.

Висновки та перспективи подальшого дослідження.

Ми розглянули відому математичну модель впливу двох взаємовиключаючих інформаційних потоків на соціум, але з введенням в неї такого поняття як запізнення в часі для одного з потоків, оскільки така ситуація є більш реальною ніж одночасний початок діяльності обох інформаційних потоків. Отримані теоретичні міркування ми спробували спроектувати на реальну ситуацію, що склалась в українському суспільстві внаслідок діяльності православних церков двох конфесій: УПЦ КП та УПЦ МП. Результати виявились надзвичайно реалістичними і такими, що потребують детальної інтерпретації з точки зору соціології. Розробкою цього питання ми і збираємось зайнятись в найближчій перспективі.

1. Д.А.Губанов, Д.А.Новиков, А.Г.Чхартишвили. Социальные сети: модели информационного влияния, управления и противоборства / Под ред. чл.-корр. РАН Д.А.Новикова. – М.: Издательство физико-математической литературы, 2010. – 228 с.
2. А.А.Самарский, А.П.Михайлов. Математическое моделирование в информационную эпоху // Вестник РАН, 2004, №9.
3. А.А.Самарский, А.П.Михайлов. Математическое моделирование. Идеи, методы, примеры. – М.: Наука, 2001, 320 с.
4. А.А.Самарский, А.П.Михайлов. Методологические основы моделирования социальных процессов: пределы возможного. Математическое моделирование социальных процессов. вып.3. – М.: МГУ, 2000.

5. А.П.Михайлов, Н.А.Маревцева, Модели информационной борьбы. Математическое моделирование, 2011, том 23, №10, С.19-32.
6. F.M.Bass. A new product growth for model consumer durables // Management Science, 1969, v.15, p.215-227.
7. Л.Л.Делицын. Количественные модели распространения нововведений в сфере информационных и телекоммуникационных технологий. – М.: МГУКИ, 2009, 106 с.
8. А.П. Михайлов, Н.В. Ключов. О свойствах простейшей математической модели распространения информационной угрозы. Математическое моделирование социальных процессов, вып.4. – М.: МАКС Пресс, 2002, с.115-123.
9. Федонюк А.А., Шигорін П.П., Антонюк Б.П. Математична модель співіснування двох релігійних конфесій в суспільстві (на прикладі України). – Математика. Інформаційні технології. Освіта // Збірник статей. – Луцьк: ПП Іванюк, 2015, с.196-206.
10. Антонюк Б.П. Модель інформаційного протистояння на прикладі розвитку мережі церковно-релігійних інституцій. – Математика. Інформаційні технології. Освіта // Збірник статей. – Луцьк: ПП Іванюк, 2016, с.5-11.

УДК 519.9

Гінайло П.І., Лісковець С.М., Тимошук В.М., Грінченко Л.Г.
Луцький національний технічний університет

НЕОБХІДНІ УМОВИ ЕКСТРЕМУМУ ДЛЯ ЛІНІЙНИХ БАГАТОЗНАЧНИХ ВІДОБРАЖЕНЬ

Гінайло П.І., Лісковець С.М., Тимошук В.М., Грінченко Л.Г. Необхідні умови екстремуму для лінійних багатозначних відображень. Отримані необхідні умови екстремуму для задачі оптимального керування, які описуються диференціальними багатозначними включеннями. При цьому враховується наявність додаткових параметрів, за допомогою яких можна впливати на функціонал і фазові обмеження задачі.

Ключові слова: екстремум, диференціальні включення, необхідні умови екстремуму, оптимальне керування, багатозначні відображення.

Гинайло П.И., Лисковец С.М., Тимошук В.Н., Гринченко Л.Г. Необходимые условия экстремума для линейных многозначных отображений. Получены необходимые условия экстремума для задачи оптимального управления, описываемые дифференциальными включениями. При этом учитывается наличие дополнительных параметров, с помощью которых можно влиять на функционал и фазовые ограничения задачи.

Ключевые слова: экстремум, дифференциальные включения, необходимые условия экстремума, оптимальное управление, многозначные отображения.

Ginaylo P. Liskovec S. Tymoshchuk V., Grinchenko L. Necessary conditions for extremum with linear multivalued mappings. Necessary extremum conditions for optimal control problems described by differential inclusions meaningful are received. This takes into account the availability of additional parameters, which can affect the functional restrictions and phase problems.

Keywords: extremum, differential inclusion necessary extremum conditions, optimal control, ambiguous map.

Постановка наукової проблеми. Різноманітні задачі оптимального керування в останні роки знаходять широке застосування в самих різних областях сучасної науки і техніки. Усі фізичні процеси, що мають місце в техніці, як правило, керовані, тобто можуть здійснюватися різними способами, в залежності від потреб людини. Тому і виникає питання про знаходження найкращого або оптимального в тому чи іншому розумінні керуванні процесом. Побудові найбільш загальних необхідних умов екстремуму присвячено дуже багато робіт різних вчених. Це умови екстремуму дають можливість передбачити структуру розв'язку.

Аналіз досліджень. Новий напрямок досліджень в теорії необхідних умов екстремуму сприяв розвитку нового підходу до багатьох задач оптимального керування, в якому центральне місце займає поняття багатозначного відображення. Справа в тому, що задачу оптимального керування виявляється зручно трактувати як задачу оптимізації на множині траєкторій деякого диференціального включення. Поняття диференціального включення дозволяє охопити багато задач оптимального керування єдиним методом розв'язання. В основі теорії диференціальних включень лежить поняття багатозначного відображення.

Метою роботи є побудова необхідних умов екстремуму для диференціальних включень при наявності додаткових обмежень на координати траєкторії в динамічних задачах з багатозначними відображеннями.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Будемо розглядати скінчені простори $X = R^n$, $Y = R^m$. Нехай a – багатозначне відображення, $a(x) \subset R^n$.

Означення 1. Будь-яку абсолютно неперервну вектор-функцію $x(t) \in R^n$ що задовольняє майже скрізь включенню

$$\dot{x}(t) \in a(x(t)), \quad t \in [0,1]$$

назвемо траєкторією.

Зафіксуємо стандартний проміжок зміни одномірному аргументу $t \in [0,1]$.

Нехай для $t \in [0,1]$ визначена абсолютно неперервна функція $x(t) \in R^n$, майже скрізь диференційована. При цьому може статися, що для майже всіх t виконується відношення

$$\frac{d}{dt} x(t) \in a(x(t)) \quad (1)$$

яке називається диференціальним включенням, тобто це диференціальне рівняння з багатозначною правою частиною.

Розв'язком диференціального включення називається абсолютно неперервна вектор-функція $x(t)$, майже скрізь задовольняє відношення (1) на заданому інтервалі змінної t .

Якщо множина $a(x)$ складається лише з одного елемента, функції $f(x)$, тобто відображення $a(x)$ – однозначне, то дане включення переходить в звичайне диференціальне рівняння

$$\frac{d}{dt}x(t) \in f(x).$$

Видно, що диференціальне включення є ніщо інше, як узагальнення диференціальних рівнянь.

До диференціальних включень приводять, наприклад, задача про функції, що задовольняють диференціальному рівнянню з заданою точністю

$$\left[\frac{dx(t)}{dt} - f(t, x(t)) \right] \leq \varepsilon$$

диференціальні нерівності

$$f\left(t, x, \frac{dx}{dt}\right) \geq 0.$$

Якщо вектор-функція $f(x, u) \in R^n$ визначена для $x \in R^n$ і $u \in U$, де U – деяка підмножина з R^s , то багатозначне відображення $a(x)$ можна визначити так

$$a(x) = f(x, u) = \{y : y = f(x, u), u \in U\}.$$

Тоді диференціальне включення (1) переходить у співвідношення

$$\frac{d}{dt}x(t) \in f(x(t), u),$$

або

$$\frac{d}{dt}x(t) \in f(x(t), u(t)), \quad u(t) \in U. \quad (2)$$

Такі рівняння (2) зазвичай зустрічаються в задачах керування. Тут $x(t)$ шукана вектор-функція, а $u = u(t)$ – керування, тобто вектор-функція, яку можна вибрати довільно серед всіх допустимих керувань.

Із вище сказаного слідує, що диференціальне включення (1) може мати множину розв'язків. В зв'язку з цим в задачах оптимального керування знаходиться оптимальний розв'язок, який задовольняє задані умови. В подальших дослідженнях при розв'язку цього питання будемо використовувати теорему 1.

Теорема 1. Якщо $a(x)$ – випуклозначне замкнуте обмежене відображення, напівнеперервне зверху по x , і $\text{dom } a$ складається з кулі радіуса

$$r = (1 + \|x_0\| e^c - 1)$$

де c – константа така, що

$$\|a(x)\| \leq c(1 + \|x\|)$$

то існує розв'язок $x(t)$ диференціального включення (1), яке задовольняє нерівність

$$\|x(t)\| \leq (1 + \|x_0\| e^{ct} - 1), \quad x(0) = x_0.$$

При цьому розв'язок $x(t)$ визначений на всьому відрізку $[0, 1]$ і задовольняє на цьому відрізку умову Ліпшиця.

Означення 2. Якщо $x(t) \in R^n$, $t \in [0, 1]$ – довільна крива, то її ε - околom називається множина точок x таких, що

$$\|x - x(t)\| \leq \varepsilon$$

при деякому $t \in [0, 1]$.

Оскільки диференціальне включення навіть при заданій початковій точці визначає не єдину траєкторію, що йому задовольняє, то має зміст ставити задачу знаходження серед усіх траєкторій диференціального включення таку, яка б володіла додатковими властивостями.

Питання про існування розв'язку оптимізаційної задачі задаватись не буде. Будемо припускати, що оптимальний розв'язок цієї задачі існує і будемо позначати його через $\tilde{x}(t)$. Тоді наша ціль заключається в тому, щоб знайти ті умови, які цей розв'язок задовольняють, тобто отримати необхідні умови екстремуму.

Сформулюємо таку задачу.

Серед усіх траєкторій $x(t)$, $t \in [0, 1]$, які задовольняють майже скрізь диференціальному включенню

$$\dot{x}(t) \in a(x(t)), \quad t \in [0, 1]$$

і умовам

$$x(0) \in N, \quad x(1) \in M$$

знайти таку, яка мінімізує вираз

$$I(x(\cdot)) = \int_0^1 g(x(t), t) dt + \varphi_0(x(1)).$$

Ця задача розв'язується при наступних умовах на дані що входять до неї. Крайні умови цієї задачі задаються множинами N і M за допомогою кінцевої системи нерівностей і рівнянь

$$N = \{x: \varphi_b(x) \leq 0\},$$

$$M = \{x: \varphi_e(x) \leq 0\}$$

де

$$\varphi_b(x) = \max \left\{ \max_{i \in I^-} \varphi_b^i(x), \max_{i \in I^0} (\varphi_b^i(x)) \right\}.$$

Означимо також для $\delta \geq 0$ для множин:

$$N_\delta = \{x: \varphi_b(x) \leq c\delta\},$$

$$M_\delta = \{x: \varphi_e(x) \leq e\delta\}$$

де $\varphi_b(x)$, $\varphi_e(x)$ - функції, які задовольняють умови Ліпшиця.

В якості функцій $\varphi_b(x)$ і $\varphi_e(x)$ можуть також бути взяті функції відстані до множини $d(x/N)$ і $d(x/M)$.

Припущення А. Множини N і M замкнуті. Множини N_δ і M_δ , $\delta \geq 0$, в кожній точці володіють локальними шатрами $K_{N_\delta}(x)$ і $K_{M_\delta}(x)$, причому якщо послідовності

$$x_k \in N_{\delta_k}, \quad x_k^* \in K_{\delta_k}^*(x_k)$$

$$x'_k \in M_{\delta_k}, \quad x'^*_k \in K_{\delta_k}^*(x'_k)$$

збігаються до x_0, x_0^* і до x_0 та x_0^* відповідно, а $\delta_k \rightarrow 0$, то

$$x_0^* \in K_N^*(x_0),$$

$$x_0^* \in K_M^*(x_0).$$

Тут в якості $K_{N_\delta}(x)$ взятий конус

$$K_{N_\delta}(x) = \{\bar{x}: \bar{x} = \gamma(x_1 - x), \quad \gamma > 0, \quad x_1 \in N_\delta\}.$$

Спряжений йому конус матиме вигляд

$$K_{N_\delta}^*(x) = \{x^* : \langle \bar{x}, x^* \rangle \geq 0, \quad \bar{x} \in K_{N_\delta}(x)\}.$$

Аналогічно

$$K_{M_\delta}(x) = \{\bar{x} : \bar{x} = \gamma(x_1 - x), \quad \gamma > 0, \quad x_1 \in M_\delta\}$$

$$K_{M_\delta}^*(x) = \{x^* : \langle \bar{x}, x^* \rangle \geq 0, \quad \bar{x} \in K_{M_\delta}(x)\}.$$

Припущення В. Функції $g(x, t)$ і $\varphi_0(x)$ неперервні по x і t та задовольняють умову Ліпшиця

$$|g(x_1, t) - g(x_2, t)| < L \|x_1 - x_2\|$$

$$|\varphi_0(x_1) - \varphi_0(x_2)| < L \|x_1 - x_2\|$$

в будь-якій обмеженій області простору X . Константа L може залежати від цієї області, але не залежить від $t \in [0, 1]$. При цьому $g(x, t)$ та $\varphi_0(x)$ допускають в кожній точці (x, t) верхню випуклу апроксимацію, а їх субдиференціали по x $\partial(g(x, t))$ і $\partial(\varphi_0(x))$ рівномірно обмежені в кожній обмеженій області і напівнеперервно зверху залежать від x і t . Припущення С. Відображення a випуклозначне і замкнуте, задовольняє умову Ліпшиця в ξ – трубці траєкторії $\tilde{x}(t)$, $t \in [0, 1]$, і множина $a(\tilde{x}(0))$ обмежена. Конуси $K_a(x, y)$ являються локальними шатрами і локально спряжене відображення $a^*(y : (x, y))$ напівнеперервне зверху залежить від своїх аргументів рівномірно обмежене для всіх x з ξ – трубки траєкторії $\tilde{x}(\cdot)$ і $y \in a(x)$. Для сформульованої задачі при зроблених припущеннях отримані необхідні умови мінімуму. Теорема 2. Нехай виконані припущення А, В, С і $\tilde{x}(\cdot)$ мінімізує функціонал

$$I(x(\cdot)) = \int_0^1 g(x(t), t) dt + \varphi_0(x(1))$$

серед всіх траєкторій, які задовольняють включення

$$\dot{x}(t) \in a(x(t)), \quad t \in [0, 1]$$

і крайні умови $x(0) \in N$, $x(1) \in M$.

Тоді існує число $\lambda_0 \geq 0$, вектор x_a^* і функція $x^*(t)$, $t \in [0, 1]$ не рівні нулю одночасно, такі що:

$$1) \quad x^*(1) + x_a^* \in \lambda_0 \partial \varphi(\tilde{x}(1)) \quad x_a^* \in K_M^*(\tilde{x}(1)), \quad x^*(0) \in K_N^*(\tilde{x}(0))$$

функція $x^*(t)$ задовольняє умову Ліпшиця

$$x^*(t) \in A^*(x^*(t); \tilde{x}(t) + \lambda \partial g(\tilde{x}(t), t))$$

майже скрізь на відрізьку $[0, 1]$

$$2) \quad \langle \tilde{x}(t), x^*(t) \rangle = W_a(\tilde{x}(t), x^*) \text{ майже скрізь на } [0, 1].$$

Приклад 2. Нехай в нашій задачі об'єкт лінійний, тобто $a(x) = Ax + U$, де U – випукла замкнута множина в R^n , A – оператор діючий з простору X в простір Y . Нехай відображення Q – багатогранне, тобто

$$gfQ = \{(x, y) : Bx - Cy \leq d\},$$

де B і C – матриці розміром $r \times n$ і $r \times m$ відповідно, а $d - r$ – мірний вектор. Тут під нерівністю для векторів розуміють систему нерівностей для компонент.

Багатозначне відображення a тут буде випуклим, замкнутим і обмеженим.

Якщо $y = Ax + \omega$, $\omega \in U$, то спряжене до a відображення буде мати вигляд

$$a^*(y^*; (x, y)) = \begin{cases} A^* y^*, & y^* \in [con(U - \omega)]^* \\ \emptyset, & y^* \notin [con(U - \omega)]^* \end{cases}$$

Якщо $(x_0, y_0) \in gfQ$, то спряжене відображення до Q обчислюється так

$$Q^*(y^*; (x_0, y_0)) = \{B^* \lambda : y^* = \lambda C^*, \lambda \geq 0, \langle Bx_0 - Cy_0 - d, \lambda \rangle \geq 0\},$$

де B^* і C^* – транспоновані матриці B і C .

Тому умова 2 теореми 2.3 в цьому випадку може бути записано наступним чином

$$-p^*(t) = \lambda(t)[u^*(t) + \alpha(t)B^*] + A^*p^*(t)$$

$$v^*(t) = \lambda(t)C$$

$$\dot{\tilde{x}} = A\tilde{x}(t) + \tilde{\omega}(t), \tilde{\omega}(t) \in U$$

$$p^*(t) \in [con(U - \tilde{\omega}(t))]^*.$$

Або, враховуючи визначення спряженого конуса і того, що

$$con(U - \tilde{\omega}(t)) = \{\lambda(\omega - \tilde{\omega}(t)) : \omega \in U, \lambda > 0\}$$

останнє відношення переписується у вигляді

$$\langle \omega - \tilde{\omega}(t), p^* \rangle \geq 0, \omega \in U.$$

Застосовуючи теорему 2.4, отримуємо

Теорема 2.8. Нехай відображення Q – багатогранне. Тоді для того, щоб траєкторія $\tilde{x}(t)$ і відповідне їй керування $\tilde{\omega}(t)$ мінімізували

$$\int_0^1 f(x(t), y(t)) dt$$

серед усіх траєкторій і керувань, які задовольняють рівняння

$$\dot{x}(t) = Ax(t) + \omega(t), \omega(t) \in U$$

крайнім умовам $x(0) \in N$, $x(1) \in M$ і умові $(x(t), y(t)) \in Q$, необхідно, щоб існувала функція $p^*(t)$, вектори $\lambda(t)$, $\alpha(t)$ і вимірні функції $u^*(t)$, $v^*(t)$ такі що:

$$p^*(0) \in K_N^*(\tilde{x}(0)), -p^*(1) \in K_M^*(\tilde{x}(1))$$

$$-p^*(t) = \lambda(t)[u^*(t) + \alpha(t)B^*] + A^*p^*(t)$$

$$v^*(t) = \lambda(t)C$$

$$\langle \omega - \tilde{\omega}(t), p^*(t) \rangle \geq 0, \omega \in U,$$

$$\lambda(t) \geq 0, \alpha(t) \geq 0,$$

$$(u^*(t), v^*(t)) \in \partial f(x(t), y(t)).$$

Висновки. Отримані необхідні умови екстремуму для задачі оптимального керування з диференціальними багатозначними включеннями. Розглянуті лінійні багатозначні відображення, при цьому враховано наявність додаткових параметрів, за допомогою яких можна впливати на функціонал і фазові обмеження задачі.

1. Пшеничный Б.Н. Выпуклый анализ и экстремальные задачи. – М.: Наука, 1980. – 320 с.
2. Пшеничный Б.Н. Необходимые условия экстремума. – М.: Наука, 1982. – 144 с.
3. Гінайло П.І. Задача оптимального керування для диференціальних включень з багатозначними відображеннями. // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2016. – № 24-25. – С. 68-72.
4. Гінайло П.І. Необхідні умови екстремуму для задачі оптимального керування з континуумом обмежень. // Наукові нотатки. Міжвузівський збірник (за напрямком „Інженерна механіка”), вип. 11, част.2, Луцьк, 2002, С.15–17.5.
Гінайло П.І. Необхідні умови екстремуму для локально випуклої задачі. // Дванадцята Міжнародна наукова конференція ім. акад. М. Кравчука. Київ, 15–17 трав. 2008 р. Матеріали конф., т. I. – К.: НТУУ “КПІ”. – 2008. – 572с.

УДК 514.181.6 + 514.182

Журило А. Г. к.т.н., доц., Сивак Є. М. к.т.н., доц.

Національний технічний університет «Харківський політехнічний інститут»

ДЕЯКІ ПИТАННЯ ЩОДО НЕВІДПОВІДНОСТІ СТАНДАРТІВ ЄСКД ТА ISO

Журило А. Г., Сивак Є. М. Деякі питання щодо невідповідності стандартів ЄСКД та ISO. У статті розглянуто невідповідності стандартів з виконання аксонометричних проєкцій за версіями ЄСКД та ISO. Така неузгодженість призводить до непорозуміння, виконання невдалих аксонометрій, зайвих побудов при кресленні аксонометричних проєкцій. Показано практичне застосування зазначених невідповідностей. Наведено приклади обмеження невідповідностей при виконання креслеників.

Ключові слова: аксонометрія, ДСТУ, ЄСКД, ISO, практичне застосування.

Журило А. Г., Сивак Е. М. Некоторые вопросы о несоответствии стандартов ЕСКД и ISO. В статье рассмотрены несоответствия стандартов по выполнению аксонометрических проекций по версиям ЕСКД и ISO. Такая несогласованность приводит к недоразумению, выполнение неудачных аксонометрий, лишних построений при вычерчивании аксонометрических проекций. Показано практическое применение указанных несоответствий. Приведены примеры ограничения несоответствий при выполнении чертежей.

Ключевые слова: аксонометрия, ГОСТ, ЕСКД, ISO, практическое применение.

Zhurilo A. G., Sivak E. M. Some questions about the inconsistency of standards of ESKD and ISO. The article discusses the inconsistencies of the standards of the axonometric projection on the version of ESKD and ISO. This inconsistency leads to misunderstanding, execution failed axonometry, unnecessary builds when drawing the axonometric projection. The practical application of these inconsistencies. Examples of constraints inconsistencies in the drawing.

Key words: axonometry, GOST, ESKD, ISO, practical use.

Постановка проблеми. Незважаючи на широкий розвиток комп'ютерної техніки та широке застосування її для виконання креслеників, появи вже декількох поколінь програм КОМПАС, AUTOCAD та їхніх аналогів, аксонометричні проєкції широко використовуються у машинобудуванні та архітектурі. Для їх опанування потрібно знати їхні властивості.

Аналіз останніх досліджень і публікацій. Питання щодо точних графічних побудов має велику історичну давнину, беручи свій початок ще в роботах Архімеда, Евкліда та інших вчених. З вичерпною повнотою і строгою науковою обґрунтованістю теорія точних метричних побудов була розроблена математиком Гаспаром Монжем, який у 1795 – 1799 рр. опублікував результати своєї двадцятирічної роботи під назвою «Нарисна геометрія» [1].

Серед імен, з якими пов'язаний розвиток наукової праці в області аксонометричних проєкцій, можна згадати видатних вітчизняних вчених: Н. М. Бескіна, О. О. Вольберга, Н. О. Глаголева, Є. А. Глазунова, А. І. Добрякова, Д. І. Каргіна, І. І. Котова, М. О. Риніна, С. О. Смирнова, М. Ф. Четверухіна [11, 12].

У даний час теорія аксонометрії розроблена докладно і висвітлена в численних працях з нарисної геометрії. Питання ж практики побудови аксонометричних зображень висвітлені в літературі недостатньо. У практиці побудови аксонометричних зображень часто виникають значні труднощі, обумовлені не тільки недостатньою підготовкою виконавця, але і складністю окремих задач, що вимагають спеціального роз'яснення [4-6, 8, 9].

Положення ускладнюється ще й тому, що за останні 20..30 років практично не публікувалося робіт із практики побудови аксонометричних зображень та її основних законів. Ті ж роботи, що були опубліковані раніше, у більшості випадків розглядають аксонометричні проєкції, не передбачені ГОСТ 2.317 – 69 або ДСТУ ISO 5456-3:2006 [2, 3].

Невирішені частини проблеми. Метою статті є доведення невідповідностей частин стандартів ДСТУ ISO та ЄСКД при виконанні аксонометричних проєкцій [7, 10].

Мета дослідження.

Забезпечення сумісного використання ДСТУ ISO та ЄСКД при виконанні аксонометричних проєкцій, та запобігання хибних побудов на креслениках при їхньому використанні.

У 2017 році Україна, у своєму прагненні до Європи, відмовилась від використання стандартів радянського періоду. Водночас з позитивними моментами у цьому є і чимало негативних моментів. Це й невідповідність стандартів один одному, і використання раніше невірних видів аксонометричних проєкцій, і той факт, що абсолютна більшість довідкової літератури, підручників, посібників, методичних вказівок стали носієм хибної інформації.

Спробуємо визначити, які невідповідності існують у ДСТУ ISO та ЄСКД, та як з

мінімальними втратами можна їх обійти.

Причому, деякі вимоги стандартів викликають здивування. Наприклад, у ДСТУ ISO 5456-3:2006 визначено, що «осі та сліди площин симетрії об'єкта можна не креслити, якщо в цьому немає потреби».

ЄСКД, навпаки, жорстко вимагає, щоб з осі симетрії об'єкта починалася його побудова, а їхня відсутність є грубою помилкою.

У ДСТУ ISO 5456-3:2006 визначено: «Штриховку, що позначає розріз або переріз, треба креслити приблизно під кутом 45° по відношенню до осей та контурів розрізу або перерізу (див. рисунок 1).

Штриховка, що позначає площини, паралельні до координатних площин, повинна бути паралельною до проєційних осей координат, як показано на рисунку 2.»

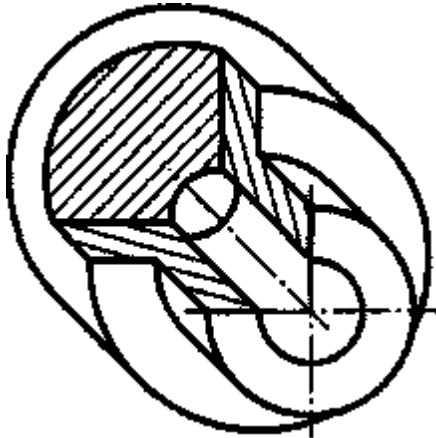


Рис. 1

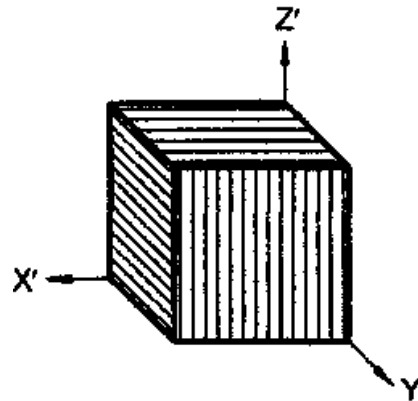


Рис. 2

Але при розгляданні рисунків 1 і 2, можна зробити висновок, що на рис. 1. штриховка виконана на перерізах, які паралельні до проєційних осей координат. Але вона виконана не так, як вказано на рис. 2.

Крім того, явно невдало обрано площини перерізу, внаслідок чого, отвір на тильній стороні деталі має невдале зображення.

Далі вказано: «Ізометрична аксонометрія надає однакової візуальної уваги усім трьом поверхням правильного шестигранника, і тому найзручніше це креслити на сітці із рівносторонніми трикутниками...». Виникає здивування, як у шестигранника (куба) автори стандарту знайшли лише три поверхні. А куди поділися інші три?

ДСТУ ISO 5456-3:2006 визначає, що: «диметричну аксонометрію використовують, коли зображуваний вид об'єкта має першочергове значення.» А чи є випадки, коли зображуваний вид об'єкта не має першочергове значення?

Для побудови симетрії ДСТУ ISO 5456-3:2006 пропонує розташування осей, що наведено на рис. 3.

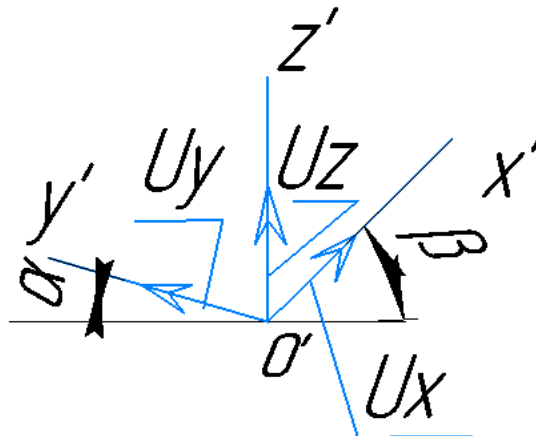


Рис. 3.

ГОСТ 2.317-69 ЄСКД «АксонOMETричні проєкції» вимагає при побудові диметричної проєкції використання іншого розташування осей.

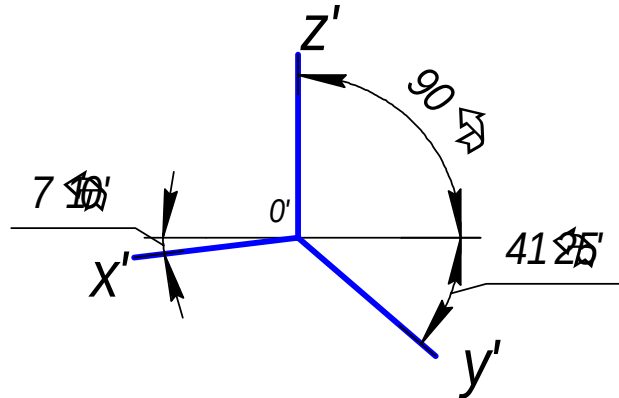


Рис. 4.

Неважко визначити, що розташування осей X та Y є дзеркальним відносно до вимог ДСТУ ISO 5456-3:2006. Це призводить до того, що, якщо спиратися лише на ГОСТ 2.317-69 ЄСКД «АксонOMETричні проєкції», то це є прямим порушенням ДСТУ ISO 5456-3:2006.

А якщо спиратися лише на ДСТУ ISO 5456-3:2006, то виявляється, що уся довідкова література, яка була видана з 1969 до 2006 рр. є хибною. Якщо звернути увагу, що лише класичний підручник з нарисної геометрії В. О. Гордона був перевиданий більш, ніж 23 рази, то зрозуміло, яку масу довідкового матеріалу треба замінити у бібліотеках ВНЗ України.

Далі, ДСТУ ISO 5456-3:2006 передбачує використання планіметричної аксонOMETрії з осями, які змінюються через кожні 15° . Незрозуміло, чим запропоновані аксонOMETрії з кутами 0° , 90° та 180° відрізняються від ортогональних проєкцій, які, як відомо, наочності не мають. А якщо звернутися до класиків аксонOMETрії, то вони визначали безліч різних аксонOMETрій, які відрізняються одна від одної кутом між осями. Інтересно, що в кожній з них виконується класична формула Риніна.

Для перевірки даної гіпотези визначимо залежність між показниками спотворення по аксонOMETричних осях і кутом φ між напрямком проєкціювання S і площиною аксонOMETричних проєкцій Π' .

Нехай задана просторова система координат X, Y, Z , на якій аксонOMETрична площина P перетинає координатні осі OX, OY, OZ відповідно у точках X, Y і Z , див. рис. 1.1.

У випадку прямокутного аксонOMETричного проєкціювання відрізок OO_p буде перпендикулярний до площини P . Тоді аксонOMETричні проєкції відрізків на осях OpX, OpY, OpZ являють собою катети прямокутних трикутників, а самі відрізки OX, OY, OZ на осях — гіпотенузи.

$$\frac{OpX}{OX} = \cos\alpha \quad \frac{OpY}{OY} = \cos\beta \quad \frac{OpZ}{OZ} = \cos\gamma.$$

Але відношення $\frac{OpX}{OX}, \frac{OpY}{OY}$ і $\frac{OpZ}{OZ}$ являють собою коефіцієнти спотворення по осі K_x, K_y і K_z . Отже, $K_x = \cos\alpha, K_y = \cos\beta$ і $K_z = \cos\gamma$.

Для відрізка OO_p косинуси кутів α_1, β_1 та γ_1 , що є додатковими до кутів α, β та γ , є напрямними косинусами. Скориставшись рис. 1.2 і деякими перетвореннями, отримаємо: $OK^2 = OK_1^2 + K_{K_1}^2$

$$OK_1^2 = y_{K_1}^2 + Oy^2 = OX^2 + Oy^2$$

$$K_{K_1} = Oz$$

$$OK_1^2 = OX^2 + Oy^2 + Oz^2$$

$$OX = OK \cdot \cos\alpha_1 \quad Oy = OK \cdot \cos\beta_1 \quad Oz = OK \cdot \cos\gamma_1$$

$$OK^2 = OK^2 \cos^2\alpha_1 + OK^2 \cdot \cos^2\beta_1 + OK^2 \cdot \cos^2\gamma_1$$

$$\cos^2\alpha_1 + \cos^2\beta_1 + \cos^2\gamma_1 = 1.$$

$$\text{Якщо } \alpha = \frac{\pi}{2} - \alpha_1, \beta = \frac{\pi}{2} - \beta_1, \gamma = \frac{\pi}{2} - \gamma_1 \quad \text{та} \quad \sin^2\alpha + \sin^2\beta + \sin^2\gamma = 1,$$

$$\text{то } 1 - \cos^2\alpha + 1 - \cos^2\beta + 1 - \cos^2\gamma = 1, \quad \cos^2\alpha + \cos^2\beta + \cos^2\gamma = 2,$$

або, іншими словами, для прямокутної аксонометричної проєкції сума квадратів коефіцієнтів спотворення дорівнює двом.

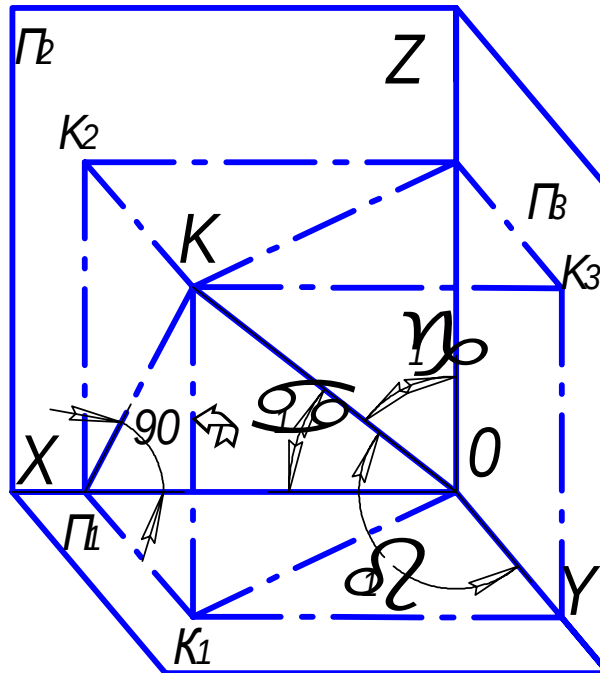


Рис. 5. Напрямні косинуси до кутів α , β та γ

З останнього рівняння видно, що коефіцієнти спотворення не можуть дорівнювати одиниці. Для ізометричної проєкції коефіцієнти спотворення по всіх осях однакові, або $\cos^2 \alpha = \cos^2 \beta = \cos^2 \gamma = a$.

$$3a^2 = 2 \quad a = \sqrt{\frac{2}{3}} \approx 0,82.$$

Відповідно при відкладанні по осях натуральних розмірів ізометричне зображення буде більше дійсного у $\frac{1}{0,82} = 1,22$ рази.

Використовуючи теорему косинусів, можна записати:

$$\begin{aligned} K_x^2 &= 1 + \frac{\cos^2 \alpha_1}{\sin^2 \varphi} - 2 \cdot \cos \alpha \frac{\cos \alpha_1}{\sin \varphi} \\ K_y^2 &= 1 + \frac{\cos^2 \beta_1}{\sin^2 \varphi} - 2 \cdot \cos \beta \frac{\cos \beta_1}{\sin \varphi} \\ K_z^2 &= 1 + \frac{\cos^2 \gamma_1}{\sin^2 \varphi} - 2 \cdot \cos \gamma \frac{\cos \gamma_1}{\sin \varphi} \\ K_x^2 + K_y^2 + K_z^2 &= \\ + \frac{\cos^2 \alpha_1 + \cos^2 \beta_1 + \cos^2 \gamma_1}{\sin^2 \varphi} - 2 \left(\frac{\cos \alpha_1 \cdot \cos \alpha + \cos \beta_1 \cdot \cos \beta + \cos \gamma_1 \cdot \cos \gamma}{\sin \varphi} \right) &= 1 + \frac{1}{\sin^2 \varphi} \end{aligned} \quad 3$$

$$K_x^2 + K_y^2 + K_z^2 = 2 + \operatorname{ctg}^2 \varphi.$$

Це відома формула М. О. Риніна, яка є основною формулою аксонометрії. При заданому напрямку проєкціювання можна довільно задаватися двома будь-якими коефіцієнтами спотворення, але величина третього з них буде визначатися наведеними рівняннями. Таким чином, при прямокутному проєкціюванні

$$K_x^2 + K_y^2 + K_z^2 = 2.$$

Або при заданому куті нахилу проєкціювання показники спотворення не є незалежними; вони зв'язані між собою деякими співвідношеннями, описаними у формулі М. О. Риніна.

Тобто, з цієї сторони, використання усіх запропонованих аксонометрій є оправданим.

Висновки та перспективи подальшого дослідження. Визначено, що існуючі стандарти далекі від досконалості, у них чимало положень, що суперечать один одному. Доведено, що в будь-якій прямокутній аксонометричній проекції повинно виконуватися положення: сума квадратів коефіцієнтів спотворення дорівнює двом.

Наочні аксонометричні зображення можуть варіюватися в дуже широких межах: від ілюзорного зображення до схематичного креслення. Тому застосування побудови різних аксонометричних проекцій залежить, насамперед, від мети і задач кожного конкретного зображення та досвіду і майстерності виконавця.

1. Гордон В. О. Курс начертательной геометрии: учебник / В. О. Гордон, М. А. Семенцов - Огиевский. – М.: Наука, 1976. – 432 с.
2. ЕСКД. ГОСТ 2.317-69 Единая система конструкторской документации. Аксонометрические проекции. М.: Издательство стандартов, 1969. – 8 с.
3. ДСТУ ISO 5456-3:2006. Кресленики технічні. Методи проєціювання. Частина 3. Аксонометричні проєкції. К.: Держспоживстандарт України, 2008. – 12 с.
4. Журило А. Г. Методика построения аксонометрических проекций тел вращения на примере изометрической проекции цилиндра / А. Г. Журило // Вестн. НТУ «ХПИ». — 2007. – № 11. – С. 78 – 81.
5. Журило А. Г. Методика построения аксонометрических проекций тел вращения на примере изометрической проекции конуса / А. Г. Журило // Вестн. НТУ «ХПИ». — 2005. – № 57. – С. 65 – 68.
6. Журило А. Г. Побудова деяких геометричних тіл у диметрії / А. Г. Журило // Вестн. НТУ «ХПИ». — 2008. – № 43. – С. 128 – 131.
7. Журило А. Г. Теоретичні та практичні основи аксонометрії [Текст] / А. Г. Журило. Навч. посібник. Х.: НТУ «ХПІ». — 2010. - 196 с.
8. Журило А. Г. Основна теорема аксонометрії – теорема Польке-Шварца та її практичне використання / А. Г. Журило, Є. М. Сивак, І. Ю. Адашевська // Комп'ютерно - інтегровані технології: освіта, наука, виробництво. — 2015. - №19. - С. 198-202. Видавництво Луцького національного технічного університету.
9. Журило А. Г. Построение аксонометрических изображений без вторичных проекций / А. Г. Журило, Е. М. Сивак, И. Ю. Адашевская // Сборник трудов XI Международной заочной конференции «Развитие науки в XXI веке» Харьков. — 2016. Ч. 1. Стр. 95-101.
10. Каменев В. И. Аксонометрические проекции : Альбом чертежей / В. И. Каменев. — Москва–Свердловск : Гос. изд - во машиностроит. лит., 1946. – 72 с.
11. Ланюк А. В. Аксонометрические проекции: учебник / А. В. Ланюк. — М. : Гос. изд - во лит - ры по строительству и архитектуре, 1956. – 176 с.
12. Порсин Ю. Я. Аксонометрические изображения машиностроительных деталей: учебник / Ю. Я. Порсин. – М.- Л. : Машгиз, 1973. – 188 с.
13. Журило А. Г. Деякі питання щодо креслення кіл при побудові аксонометричних проєкцій / А. Г. Журило, Є. М. Сивак // Комп'ютерно - інтегровані технології: освіта, наука, виробництво. — 2017. - №26. - С. 93-98. Видавництво Луцького національного технічного університету.

УДК 622.831.2:539.422.3

Л.М.Захарова

Інститут фізики гірничих процесів НАН України

РОЗРОБКА АЛГОРИТМУ ІДЕНТИФІКАЦІЇ ДИСИПАТИВНИХ СТРУКТУР Й ВИЗНАЧЕННЯ ЇХ ПАТТЕРНІВ

Захарова Л.М. Розробка алгоритму ідентифікації дисипативних структур й визначення їх паттернів.

Обґрунтована методологія ідентифікації дисипативних структур, що виникають у процесі необоротних зрушень масиву гірських порід навколо гірничих підземних виробок. Розроблена методика кластеризації паттернів дисипативних структур з урахуванням невизначеності й варіації гірничо-геологічних й гірничотехнічних умов розробки родовищ корисних копалин

Ключові слова: необоротні зрушення, дисипативні структури, кластеризація.

Захарова Л.М. Разработка алгоритма идентификации диссипативных структур и определение их паттернов. Обоснована методология идентификации диссипативных структур, возникающих в процессе необратимых сдвижений массива горных пород вокруг горных подземных выработок. Разработана методика кластеризации паттернов диссипативных структур с учетом неопределенности и вариации горно-геологических и горнотехнических условий разработки месторождений полезных ископаемых.

Ключевые слова: необратимые сдвижения, диссипативные структуры, кластеризация.

Zakharova L.M. Development of an algorithm for dissipative structures identification and determination of their patterns. The objective of this paper is to develop a methodology for dissipative structures identification and determination of their position in space and in time. We used K-mean clustering to define coordinates of the cluster centers and Voronoi diagrams for positioning their boundaries. Boltzmann formula has been used for identification of the optimal moment when clustering process should be terminated. We found that reliable identification of the dissipative structures in space and in time can be done by monitoring of incremental rock mass displacement in the diapason from 2 to 10 standard error of measurement. Coordinates of the cluster centers can be positioned with variogram method using K-means technique and entropy minimization principle. It was the first time when cluster boundaries and their number was defined for dissipative structure, which spontaneously occur during irreversible ground movement in uncertain geologic environment. Practical implication of this methodology facilitates safety maintenance during underground deposits exploitation.

Key words: irreversible ground movement, dissipative structures, clustering.

Вступ. Проблема розпізнавання образів на сьогодні стосується вельми широкого кола галузей знань: від розпізнавання мовлення до геоінформаційних технологій і робототехніки [1,2]. Однією з важливих ланок процесу розпізнавання є кластеризація певних паттернів, які зазвичай змінюються як у просторі так і в часі [3]. Найбільш складна ситуація виникає у випадках, коли у процесі розпізнавання необхідно враховувати особливості об'єкта та його еволюцію. Наприклад під час необоротних зрушень масиву гірських порід навколо підземних виробок спонтанно виникають дисипативні структури, паттерни яких еволюціонують так, щоб накопичувати максимальний об'єм степеней вільності [4,5]. Причому розмір дисипативних структур та їх паттерни змінюються у часі нелінійно, що ускладнює процес їх розпізнавання. Додатково процес еволюціонування дисипативних структур відбувається на тлі шумів, що породжуються випадковими флуктуаціями фізико-механічних властивостей порід, невизначеністю граничних умов, а також параметрів технологій розробки родовищ корисних копалин. Тому невирішеною частиною проблеми залишається не тільки знаходження положення границь кластерів у просторі, але й визначення оптимальних параметрів методики кластеризації, зокрема моментів, у які необхідно виконувати зйомку дисипативних структур [5].

Отже метою даної статті є розробка алгоритму ідентифікації дисипативних структур у масиві гірських порід під час їх необоротних зрушень навколо підземних виробок й визначення паттернів дисипативних структур у процесі їх еволюції.

Викладення основного матеріалу. Графік (а) рис. 1 показує, що збільшення похибки вимірювань зменшує ймовірність виявлення необоротних інкрементальних зрушень. Низька точність фіксації координат міток призведе до втрати окремих або самих малих, проте закономірних інкрементів переміщень міток та геодезичних знаків. Ці інкременти є основним термодинамічним представником (поток) необоротного процесу, який може породжувати дисипативні структури. Часовий інтервал між послідовними сесіями моніторингу є наступним вирішальним параметром поруч зі щільністю розміщення міток на ділянці, де виконуються інструментальні спостереження. З одного боку збільшення вказаного інтервалу впливає на достовірність виявлення інкрементів зрушень неоднозначно. Спочатку збільшення часового інтервалу сприяє підвищенню достовірності результатів моніторингу, оскільки зменшується

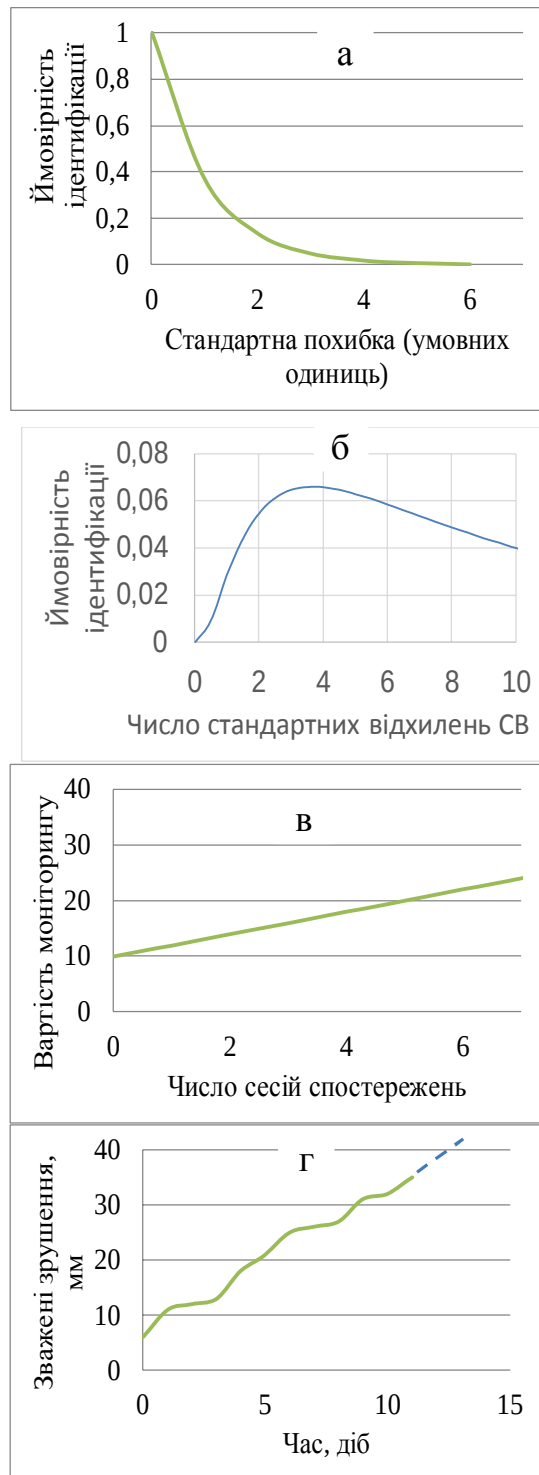


Рис. 1. Графіки, що ілюструють процес вибору ідентифікації дисипативних структур: а, б, в і г – відповідно вибір стандартної похибки вимірювань, числа стандартних відхилень, числа сесій моніторингу й його часового інтервалу

розмірів блоків. У випадку мінімальної відстані моніторинг здатен уловлювати особливості необоротного руху кожного з блоків. Зі зростанням середньої відстані, вираженої у величинах характерного розміру блочності інформація про дисипативні структури буде ставати більш

кількість хибних інкрементальних зрушень, які зафіксовані як випадкова похибка вимірювань, а не закономірне необоротне зрушення, яке відбулось у реальності (фрагмент б рис. 1). З другого боку подальше збільшення часового інтервалу призведе до зростання середньої величини інкрементальних зміщень, що викликає втрату цінної інформації й тягне за собою спрощення загальної картини необоротних зрушень. Це безпосередньо погіршить достовірність виявлення дисипативних структур.

Варто також мати на увазі економічну сторону процесу моніторингу координат міток. Вартість моніторингу зростає як зі збільшенням щільності міток у просторі, так й зі зменшенням інтервалу між послідовними сесіями моніторингу. На фрагменті (в) точка перетину осі ординат з графіком вартості характеризує одноразові початкові витрати на обладнання спостережної мережі міток й геодезичних знаків. Чим більше щільність геодезичних знаків (тобто менше середня відстань між ними), тим вища вартість побудови геодезичної мережі. Зрозуміло також, що зі скороченням часового інтервалу між послідовними сесіями моніторингу координат міток збільшується число сесій, що прямо пропорційно підвищує вартість моніторингу (поточні витрати).

Отже проблема обґрунтування точності вимірювання, щільності установки а також часового інтервалу між послідовними сесіями є складною оптимізаційною задачею, яка повинна вирішуватись окремо. При виконанні даної роботи раціональні параметри моніторингу обґрунтовані експериментальним шляхом. Перш за все необхідно визначитись, якого типу структури необхідно ідентифікувати: мікро-, макро-, чи мега- згідно класифікації ІГТМ. У цій роботі основна увага була присвячена виявленню дисипативних структур у макро-масштабі, який є співрозмірним з габаритами гірничих виробок, або розмірами локальних зсувів ґрунтів на похилій земній поверхні.

Перш за все необхідно вибрати раціональну відстань між мітками або реперами, мережа яких буде покривати площу, яка охоплена процесом необоротних зрушень масиву гірських порід, або ґрунтів. Ця відстань визначається ключовим структурним елементом, який являє собою окремий блок породи або ґрунту. Практика свідчить про те, що для адекватного відображення поведінки дискретного або блочного середовища достатньо об'єму, розміри якого лише на порядок перевищують характерні розміри породних блоків як складових масиву [6]. Отже типова відстань між геодезичними знаками може знаходитись у межах від 2 до 10 характерних

узагальненою і деякі деталі буде втрачено, проте в інтервалі відстаней до 10 характерних розмірів блоків надійна ідентифікація дисипативних структур гарантована.

Мінімізація вартості моніторингу еволюції дисипативних структур досягається методом вибірки групи ділянок масиву, які представляють на поточний момент найбільший інтерес. Подальша оптимізація плану моніторингу досягається елементами управління проектами а також застосування рівнів ризику з урахуванням невизначеності подальшого розвитку небезпечних процесів. Наприклад задаючи певний рівень ризику, можна обмежитись лише тими ділянками, на яких на поточний момент розвиваються процеси, які являють максимальний ризик катастрофи й планувати частоту спостережень саме виходячи з цих процесів [5].

До речі невизначеність процесів руйнування й необоротного деформування та зрушення порід і твердого тіла взагалі є його фундаментальною властивістю. Тому вибір часового інтервалу моніторингу необхідно планувати згідно поточним темпам розвитку необоротних процесів (зрушень) з урахуванням їх невизначеності (фрагмент (г) рис. 1, де прогноз показаний пунктиром.

У даній роботі для виявлення дисипативних структур застосовуються сучасні потужні методи варіаграмного аналізу [7] а також метод К-середніх [8,9]. Варіограма являє собою статистичний момент другого порядку, який широко застосовується у геостатистиці для моделювання й аналізу просторової кореляції.

Варіограма $2\gamma(x, x+h)$ для значень просторової змінної $Z(x)$ у двох точках x і $x+h$, які розділені вектором h , визначається варіацією різниці значень змінної у вказаних точках:

$$2\gamma(x, x+h) = E[(Z(x) - \mu(x)) - (Z(x+h) - \mu(x+h))]^2 \quad (1)$$

На рис. 2 наведено вид варіограми. Існує кілька базових типів варіограм. У даному випадку варіограма схожа на експоненціальний тип. За формою цієї варіограми можна зробити кілька важливих висновків. Оскільки крива, що апроксимує варіограму має тенденцію пересікати вісь ординат, усі внутрішні закономірності варіабельності статистичних даних (а в даному випадку довжин векторів інкрементальних зрушень і їх напрямків) варіограмний аналіз уловив.

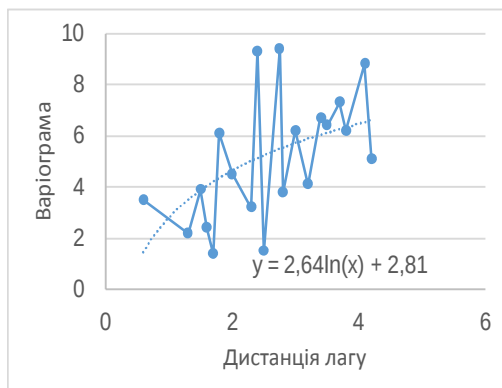


Рис. 2. Приклад варіограми

Варіограма може мати поріг, проте у даному випадку важливо, що параметри більш близьких векторів корелюють між собою краще, ніж віддалених, що помітно по зростанню середніх значень варіограми з віддаленням (збільшенням довжини вектору h) між парами точок. До того ж суттєві коливання значень варіограми відносно кривої, яка її інтерполіує свідчить про наявність певних структур у межах групи векторів. Це означає високу ймовірність того, що вказана група векторів складається з кластерів, які характеризують різні дисипативні структури.

Для ідентифікації вказаних структур застосовано метод К-середніх. Ідея методу заснована на одночасній мінімізації варіації відстаней між векторами всередині кластерів і максимізації розкиду відстаней між центрами

тяжіння кластерів. Зауважимо, що відстані між векторами визначаються у Евклідовому просторі, який має довільну розмірність. Задача класифікації випадкових величин, зокрема для виявлення дисипативних структур серед векторної мозаїки інкрементальних необоротних зміщень масиву гірських порід, не має однозначного рішення. Тому кількість можливих кластерів змінювалося від 3-4 і далі. Процес класифікації контролювався динамікою дисперсій або середньоквадратичних відхилень відстаней від центрів кластерів до спільного центру. Важливо підкреслити, що для візуалізації кластерної мозаїки використовувалося інтерпольоване поле зсувів ґрунту, тоді як у процесі кластерного аналізу залучалися тільки фактичні зміни координат реперів, установлених на експериментальній ділянці по мітках.

У таблиці 1 показані результати обчислення відстаней між кластерами у Евклідовому просторі на етапі, коли їх було сім штук. У міру збільшення числа кластерів довжина і орієнтація векторів зсувів їх центрів змінювалася. Деякі вектори демонстрували відносну стабільність, тоді як ряд векторів розпадалися на складові, що свідчило про різноманіття кластерної мозаїки і необхідність подальшого ітераційного процесу вичленування кластерів.

У більшості випадків на початковому етапі кластеризації всі вектори необоротних зрушень міняли свої параметри істотно. Так у процесі збільшення числа векторів з 5 до 7 відносно

стабільним залишився лише вектор. Решта векторів змінили або довжину, або поміняли як довжину, так і напрямок.

Таблиця 1.

Відстань між кластерами, м ² \ м *							
№ кластеру	1	2	3	4	5	6	7
1	0,0	2625,8	7398,7	1921,9	3231,4	8435,1	7990,9
2	51,2	0,0	2983,1	2990,7	3069,5	1672,4	2321,8
3	86,0	54,6	0,0	2976,9	12058,5	2567,0	8194,8
4	43,8	54,7	54,6	0,0	8432,0	7034,2	10574,4
5	56,8	55,4	109,8	91,8	0,0	7337,3	3031,1
6	91,8	40,9	50,7	83,9	85,7	0,0	2094,6
7	89,4	48,2	90,5	102,8	55,1	45,8	0,0

* Вище діагоналі вказані квадрати відстаней, нижче – відстані

Процес насичення дисперсій відстаней між кластерами векторів переміщень порід і всередині кластерів наочно ілюструється графіками на рис. 3. На цьому рисунку показана дисперсія відстаней між векторами зрушень всередині кластерів і між центрами окремих кластерів. При цьому розкид відстаней вказано окремо за компонентами зрушень уздовж осей X і Y. Підкреслимо, що кластеризація векторів здійснювалася окремо по їх компонентам, оскільки важлива не тільки довжина векторів, а й їх напрямок



Рис. 3. Стабілізація дисперсії від кількості кластерів

Як видно з графіків, при збільшенні числа кластерів до 9 спостерігається стабілізація відповідних дисперсій по обидвом компонентам зрушень точок земної поверхні. Разом з тим подальше продовження процесу кластеризації показало, що по мірі зростання числа кластерів алгоритм К-середніх уловлює все тонкіше різницю між кластерами [5]. Очевидно, що у певний момент часу алгоритм почне класифікувати векторну мозаїку за ознаками, які мають не фізичну, а стохастичну

природу. Це свідчить про те, що для вибору моменту зупинки зазначеного процесу необхідний додатковий критерій.

Виходячи з другого закону термодинаміки [10], нами було доведено, що найбільш ймовірний стан товщі гірських порід, ґрунту або наносів відповідає максимуму ентропії, яка виражається за допомогою відомої формули Больцмана через ймовірність виникнення кожного кластера p_j

$$S_i = -k \sum p_j \ln p_j, \quad 1 \leq j \leq N, \quad (2)$$

де k – стала.

При цьому ентропія S_i досягає мінімуму, коли ймовірності виникнення всіх кластерів рівні між собою, тобто:

$$p_j = 1 / N \quad (3)$$

У такому випадку $dS_i / dt = 0$.

В реальних умовах результати спостережень завжди мають певну похибку і тому умова (3) недосяжна, однак вона є тією величиною, на яку можна нормувати фактично отримані ймовірності виникнення кластерів в процесі виконання кластеризації.

Таким чином, у процесі кластеризації поля необоротних зрушень ґрунтової товщі підраховувалася кількість випадків, коли з точністю до похибки вимірювань виникав один і той же кластер при збільшенні їх числа. При цьому обчислюється ентропія дисипативної системи згідно залежності (2) через ймовірність p_j , яка розраховувалася як частка повторюваного кластера у загальній серед них кількості на поточному етапі кластеризації. Ця величина нормувалася на мінімально можливу ентропію. Результати експерименту показали, що дійсно, існує момент, коли ентропія досягає мінімуму, за яким слід зупинити процес кластеризації.

Друга проблема стосувалась визначення місця розташування границь кластерів. Справа в тому, що окремі члени кластера можуть бути розкидані в просторі так, що границі кластерів перекриваються, що вносить помітну невизначеність у кінцевий результат. У зв'язку з цим була використана діаграма Вороного [11], яка визначає положення кордонів кластерів на лініях, які перпендикулярні центрам відрізків, що з'єднують центри тяжкості сусідніх кластерів. Саме таке положення границь відповідає умові (3), що мінімізує невизначеність, обумовлену похибкою визначення координат реперів і точок на місцевості, або міток на моделі.

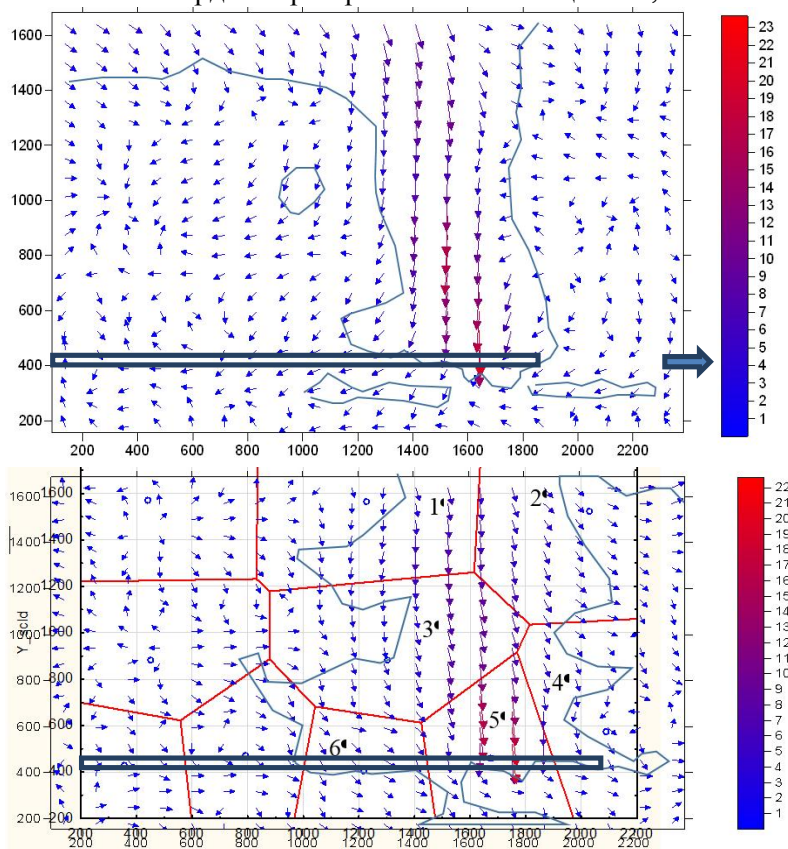


Рис. 4. Приклад кластеризації паттернів дисипативних структур у масиві гірських порід, під час відпрацювання очисного вибію

На рис. 4 показано поєднаний розподіл векторів зрушень масиву гірських порід і мозаїки Вороного. Значимі кластери, що визначені у Евклідовому просторі з достовірністю не менш за 90% помічені номерами від 1 до 6. Підкреслимо, що визначення відстані між векторами у Евклідовому просторі означає, що вказані відстані обчислюються за всіма параметрами векторів, тобто напрямком й амплітуді. Це означає, що певний кластер на перший погляд може містити різні (за довжиною, чи напрямком) вектори. Проте метод К-середніх гарантує, що усередні кластера різниця між векторами буде мінімальною за усіма параметрами. Аналіз показує, що розроблена методика кластеризації дисипативних структур досить ефективна, оскільки розділення на кластери уловлює фізичний смисл процесів необоротних зрушень товщі гірських порід. Так кластери 1 і 2 розміщені поза так

званою зоною повних обвалень, яка характеризується максимальною дезінтеграцією породної товщі і має виражену блочність.

Можна сказати, що кластери 1 і 2 розташовані вже у зони плавних прогинів товщі, яка підроблюється очисним вибієм. Різниця між кластерами 1 і 2 полягає у тому, що вектори кластеру 2 явно повернуті у бік посування очисного вибію, тоді як вектори кластеру 1 орієнтовані униз, або мають тенденцію скошуватись у бік виробленого простору.

Кластер 4 розташований над крайовою частиною вугільного пласта у так званій зоні максимального опорного тиску. Кластер 5 охоплює зону найбільш інтенсивних зрушень, яка розташована над робочим простором очисного вибію. Кластери 3 і 6 зміщені у вироблений

простір, де обвалені породи починають ущільнюватись, проте вектори кластеру 6 мають помітно більшу компоненту зрушень у напрямку посування лави, ніж вектори кластеру 3.

Отже розроблена методика кластеризації векторів інкрементальних зрушень дозволяє надійно ідентифікувати дисипативні структури, локалізуючи не тільки їх границі, але й враховуючи особливості фізичних необоротних процесів, що формують вказані структури.

Висновки. Обґрунтована методологія ідентифікації дисипативних структур, яка включає вибір оптимальної відстані між мітками або геодезичними знаками на місцевості, необхідну точність вимірювань їх координат й раціональний інтервал між послідовними сесіями моніторингу. Додатково запропоновані прийоми і заходи, які знижують вартість й трудомісткість моніторингу, а також забезпечують ідентифікацію дисипативних структур наперед заданого типу. Ефективне виявлення дисипативних структур забезпечується завдяки тому, що з одного боку параметри моніторингу гарантують ухилення від випадкових флуктуацій зміщень, обумовлених похибкою вимірювань, а з другого – унеможлиблюється пропуск інкрементальних необоротних зрушень, які несуть важливу інформацію про шлях навантаження системи.

Надійність розробленої методики ідентифікації дисипативних структур перевірена й підтверджена на фізичній моделі з дискретних матеріалів. Розроблена методика кластеризації дисипативних структур методами варіограм і К-середніх, причому границі кластерів встановлюються за допомогою діаграм Вороного. На основі формули Больцмана для ентропії обґрунтовано термодинамічний критерій оптимальної зупинки процесу кластеризації, що забезпечує достовірність виявлення дисипативних структур на рівні не менше за 90%.

Таким чином дисипативні структури у масиві гірських порід після його переходу у позамежний стан надійно виявляються моніторингом його інкрементальних (елементарних) зрушень, величина яких знаходиться у межах 2-10 середньо-квадратичних похибок вимірювання зрушень, причому центри кластерів визначаються методом варіограм і К-середніх, оптимальне число кластерів фіксується у момент досягнення ентропійного мінімуму, а їх межі локалізуються полігонами Вороного у Евклідовому просторі, що дозволяє виявити вказані дисипативні структури.

1. Pattern Recognition and Image Analysis 8th Iberian Conference, Alexandre L.A. (Ed.) IbPRIA, 2017, Faro, Portugal, June 20-23, 2017, Proceedings. 496 p.
2. Ramachandra, T.V., Bharath, H. A. and Sowmyashree, M. V. Analysis of spatial patterns of urbanisation using geoinformatics and spatial metrics. Theoretical and empirical research in urban management. 2013, 8(4): 4-24.
3. Яне Б. Цифровая обработка изображений, Техносфера, 2007, 596 с.
4. Nazimko I.V., Zakharova L.M. Clustering of the Ground Movement using Photogrammetric Methods. Proc. Int. Cconf. "Surveying in geotechnologies". NMU, Dnipropetrovsk, 2010: 46-52.
5. Zakharova, L. Generating of dissipative structures during ground irreversible movement. Transactions of Kremenchuk Mykhailo Ostrohradskyi National University, 2017, 8(105), 118-122.
6. Zhao, L.Y., Zhu, Q.Z., Xu, W.Y., Dai, F., & Shao, J.F. A unified micromechanics-based damage model for instantaneous and time-dependent behaviors of brittle rocks. International Journal of Rock Mechanics and Mining Sciences, 2016, 84, 187-196.
7. Berikov V., Pestunov I. Ensemble clustering based on weighted co-association matrices: Error bound and convergence properties. Pattern Recognition, 2017, 63(3): 427-436.
8. Gonzalez S., Garcia S., Lazaro M., Figueiras-Vidal A.R., Herrera F.. Class Switching according to Nearest Enemy Distance for learning from highly imbalanced data-sets. Pattern Recognition, 2017, 70(10): 12-24.
9. Rodrigues E.O., Torok L., Panos Liatsis, Jos. Viterbo, Aura Conci .k-MS: A novel clustering algorithm based on morphological reconstruction Pattern Recognition, 66(7) 2017: 392-403.
10. Kondepudi, D, & Prigogin, I. Modern Thermodynamics: from Heat Engines to Dissipative Structures. Second edition. John Wiley & Sons; XXVI. 2015.
11. Voronoi G.F. Nouvelles applications des parametres continus la theorie de formes quadratiques. Journal fer die reine und angewandte Mathematik. 1908, 134: 198-287.

УДК 515.2

Ісмаїлова Н.П., д.т.н., доц., Трушков Г.В.
Військова академія (м. Одеса)

ГЕОМЕТРИЧНЕ МОДЕЛЮВАННЯ ПРОСТОРОВОГО ПАРАМЕТРИЧНОГО КІНЕМАТИЧНОГО ГВИНТА

Ісмаїлова Н.П., Трушков Г.В. Геометричне моделювання просторового параметричного кінематичного гвинта. Розглянуто алгоритм формування параметричного кінематичного гвинта і його геометрична модель в 3d. Створена геометрична підпрограма на базі програми САПР Autodesk Inventor, яка реалізує автоматичну побудову діаграми просторового параметричного кінематичного гвинта за заданими вихідними параметрами в системі САПР Autodesk Inventor.

Ключові слова: спряжені поверхні, параметричний кінематичний гвинт, система САПР Autodesk Inventor, геометрична модель, аксоїди, гелікоїди.

Исмаилова Н.П., Трушков Г.В. Геометрическое моделирование пространственного параметрического кинематического винта. Рассмотрен алгоритм формирования параметрического кинематического винта и его геометрическая модель в 3d. Создана геометрическая подпрограмма на базе программы САПР Autodesk Inventor, которая реализует автоматизированное построение диаграммы пространственного параметрического кинематического винта по заданным исходным параметрам в системе САПР Autodesk Inventor.

Ключевые слова: сопрягающие поверхности, параметрический кинематический винт, система САПР Autodesk Inventor, геометрическая модель, аксоиды, геликоиды.

Ismailova N.P., Trushkov G.V. Geometric modeling of the spatial parametric kinematic screw. The algorithm for forming a parametric kinematic screw and its geometric model in 3d are considered. A geometric subroutine was created for the ADS Autodesk Inventor program, which implements an automated construction of a spatial parametric kinematic screw diagram for given initial parameters in the system ADS Autodesk Inventor.

Keywords: conjugating surfaces, parametric kinematic screw, system ADS AutodeskInventor, geometric model, axoids, helicoids.

Постановка наукової проблеми.

Розробити моделювання геометричної моделі просторового кінематичного параметричного гвинта за допомогою сучасних комп'ютерних технологій для обробки гвинтових нелінійчатих поверхонь складного профілю стосовно к зубчастим зачепленням. Метою даного дослідження є геометричне 3d - моделювання діаграми просторового кінематичного гвинта, яка дозволяє підвищити продуктивність розрахунково-конструкторських робіт і точність профілювання зубчастого зачеплення.

Аналіз досліджень.

При профілюванні спряжених поверхонь деталей в машинобудуванні застосовувались трохоїдографи та електронно-моделюючі установки [1], які мали низьку точність та продуктивність побудови сімейства трохоїд.

Кінематичний гвинт добре себе зарекомендував для формування спряжених лінійчатих поверхонь [3], тому встало актуальне завдання геометричне 3d - моделювання діаграми кінематичного параметричного гвинта для формування спряжених нелінійчатих поверхонь. Методика геометричне 3d - моделювання діаграми просторового параметричного кінематичного гвинта заснована на теоремах професора А.М. Подкоритова.

Основна частина.

Найбільш складною задачею є визначення характеристик сімейств спряжених гвинтових аксоїдів. З них розглянемо пару найбільш загального випадку спряжених гвинтових аксоїдів (рис 1.).

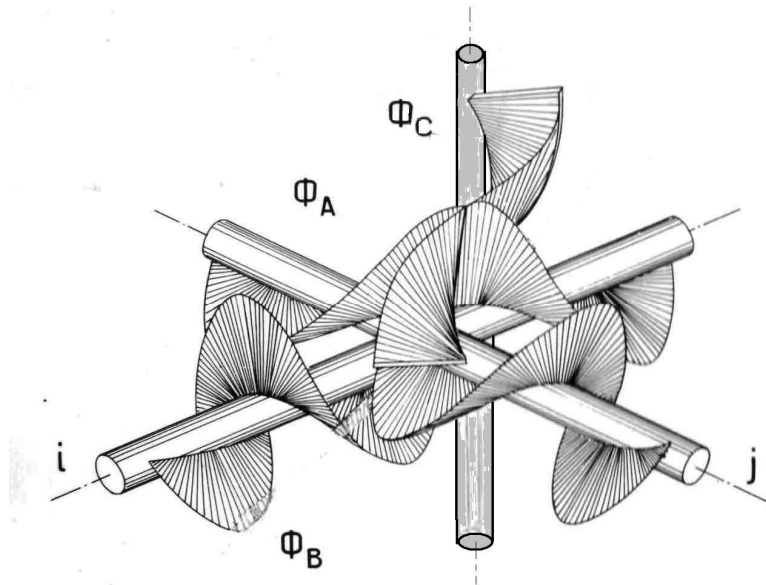


Рис. 1. Пара спряжених аксоїдів.

Метою є дослідження спряжених поверхонь, з яких обидві є гвинтовими. В основі цього процесу лежить сукупність двох рухів: обертального та поступового, що направлений по вісі обертання.

Усі точки оригіналу під час гвинтового руху описують гвинтові паралелі, що мають загальну вісь. Кінематичний зміст побудови діаграми гвинта в 3d-модельованні полягає в тому, що гвинтові лінійчаті поверхні є спряженими. Лінія дотику є похідна, навколо якої проходить обертання з кутовою швидкістю, що ковзають уздовж неї зі швидкістю даних гвинтових поверхонь в їх відносному русі одного щодо іншого.

Таким чином, діаграма гвинта дозволяє наочно уявити складання і розкладання руху твердого тіла, знаходити характеристики поверхонь в гвинтовому русі і визначити спряженні нелінійчаті поверхні з нелінійним контактом.

Сутність кінематичного методу профілювання спряжених поверхонь Σ_A та Σ_B для найбільш загального випадку, при якому, як поверхня, що утворюється Σ_A , так і Σ_B , що її огинає, є гвинтовими нелінійчатыми поверхнями з різними гвинтовими параметрами.

Геометричне 3d - модельовання діаграми просторового кінематичного параметричного гвинта [3] на базі сучасних комп'ютерних технологій, дозволить визначити параметри кінематичного гвинта без аналітичних розрахунків, котрі дозволять геометрично і наочно визначити спряженні нелінійчаті поверхні з нелінійним контактом.

Параметричний кінематичний гвинт – це просторове зображення трьох гвинтових рухів, з яких два є гвинтовими, а третє - результируючим [1].

При формуванні тривимірної моделі просторового параметричного кінематичного гвинта створена та відредагована таблиця параметрів тривимірної моделі (рис. 2), в якій можна змінювати значення, вводити параметри. При цьому кожен параметр моделі набуває значення відповідного призначеного для користувача параметра, зазначеного в колонці «Формула».

При розробці підпрограми до САПР Autodesk Inventor із формування тривимірної моделі кінематичного гвинта, створена тривимірна модель може бути використана в якості шаблону. За допомогою 13 взаємопов'язаних параметрів побудуємо в системі САПР Autodesk Inventor:

Задані параметри

1. АВ - відстань між перехресними вісями і та j (АВ = 226,64 мм);
2. f - передавальні відношення ($f = a/b$);
3. γ - кут між вісями і та j ($\gamma = 121^\circ$);
4. h_1 - крок гелікоїда Σ_A ($h_1 = AC = 80,93$ мм);
5. h_2 - крок гелікоїда Σ_B ($h_2 = BD = 56,08$ мм);
6. ω_A - швидкість обертання осі і ($\omega_A = 54,34$ мм);
7. ω_B - швидкість обертання осі j ($\omega_B = 58,92$ мм);

Отримано автоматично в результаті побудови в системі САПР Autodesk Inventor:

1. a - відстань між перехресними вісями i та m ($a = AK = 119,012$ мм);
2. b - відстань між перехресними вісями j та m ($b = BK = 107,628$ мм);
3. α - кут між вісями i та m ($\alpha = 54,53^\circ$);
4. β - кут між вісями j та m ($\beta = 66,47^\circ$);
5. h_m - крок гелікоїда Φ ($h_m = KK' = 131,927$ мм);
6. ω_m - швидкість обертання осі m ($\omega_m = 55,086$ мм).

при цьому вихідними є наступні параметри: AB , γ , ω_A , ω_B , h_1 і h_2 .

Имя параметра	Единица	Формула	Номин. зна.	Доп.	Значение в	Клк	Примечание
Параметры модели							
d3	мм	AB	226,640000		226,640000		
d4	мм	b	107,628000		107,628000		
d6	мм	k	131,927000		131,927000		
d7	мм	h2	56,080000		56,080000		
d8	мм	h1	80,930000		80,930000		
d9	град	0 град	0,000000		0,000000		
d10	град	0 град	0,000000		0,000000		
d11	град	0 град	0,000000		0,000000		
d12	град	0 град	0,000000		0,000000		
d13	мм	h2	56,080000		56,080000		
d14	град	beta	54,530000		54,530000		
d15	мм	h1	80,930000		80,930000		
d16	град	alpha	66,470000		66,470000		
d17	мм	AB	226,640000		226,640000		
d18	мм	b	107,628000		107,628000		
d20	мм	k	131,927000		131,927000		
d21	мм	h2	56,080000		56,080000		
d22	мм	h1	80,930000		80,930000		
Ссылочные параметры							
a	мм	119,012 мм	119,012000		119,012000		
d19	мм	119,012 мм	119,012000		119,012000		
Пользовательские							
AB	мм	226,64 мм	226,640000		226,640000		
b	мм	107,628 мм	107,628000		107,628000		
k	мм	131,927 мм	131,927000		131,927000		
h2	мм	56,08 мм	56,080000		56,080000		
h1	мм	80,93 мм	80,930000		80,930000		
kr	бр	a / b	1,105772		1,105772		
beta	град	54,53 град	54,530000		54,530000		
alpha	град	66,47 град	66,470000		66,470000		

Рис.2. Таблица с выходными параметрами

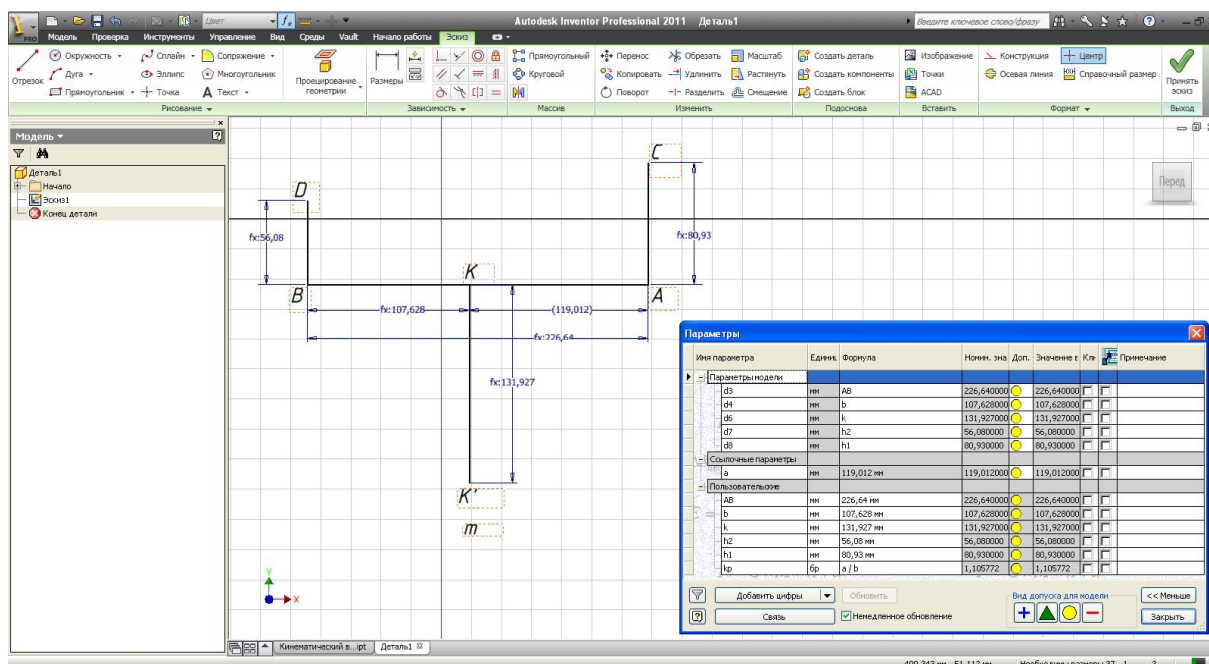


Рис.3. «Ескіз 1». Створена «Робоча площина 1» в площині «Ескізи 1»

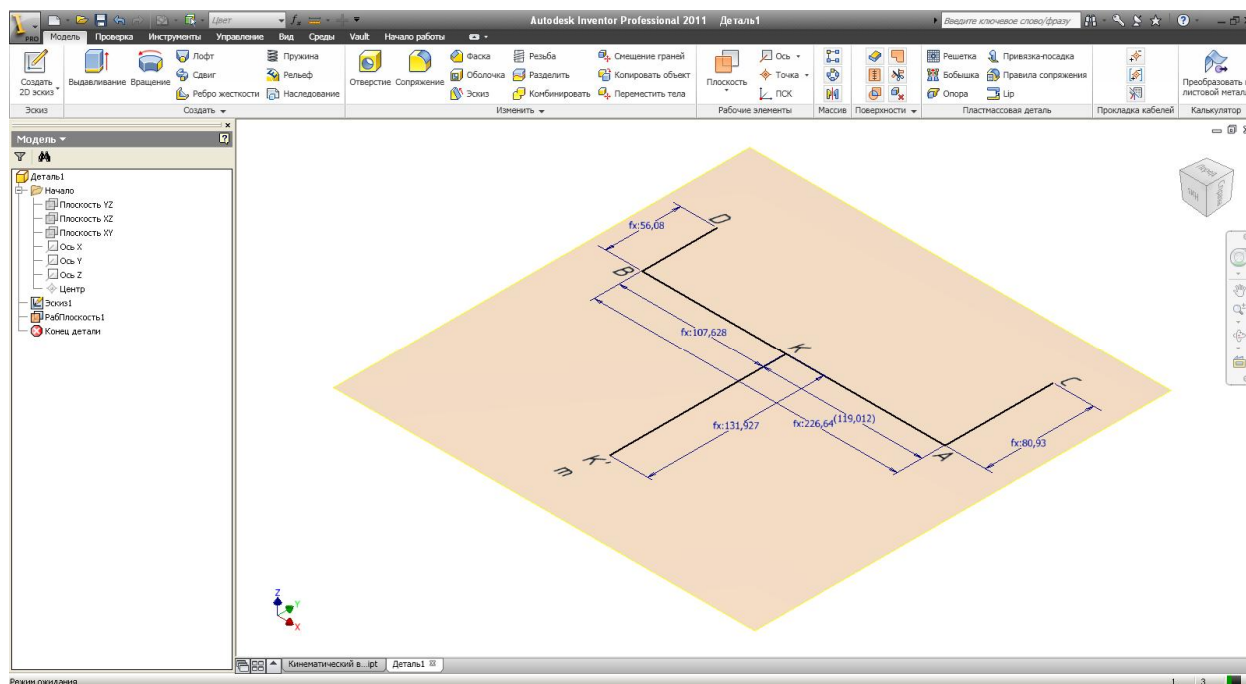


Рис.4. «Робоча площина 1»

Перпендикулярно до «Робочої площини 1» створена «Робоча площина 2», що проходить через відрізок BD «Ескізи 1» (рис.5).

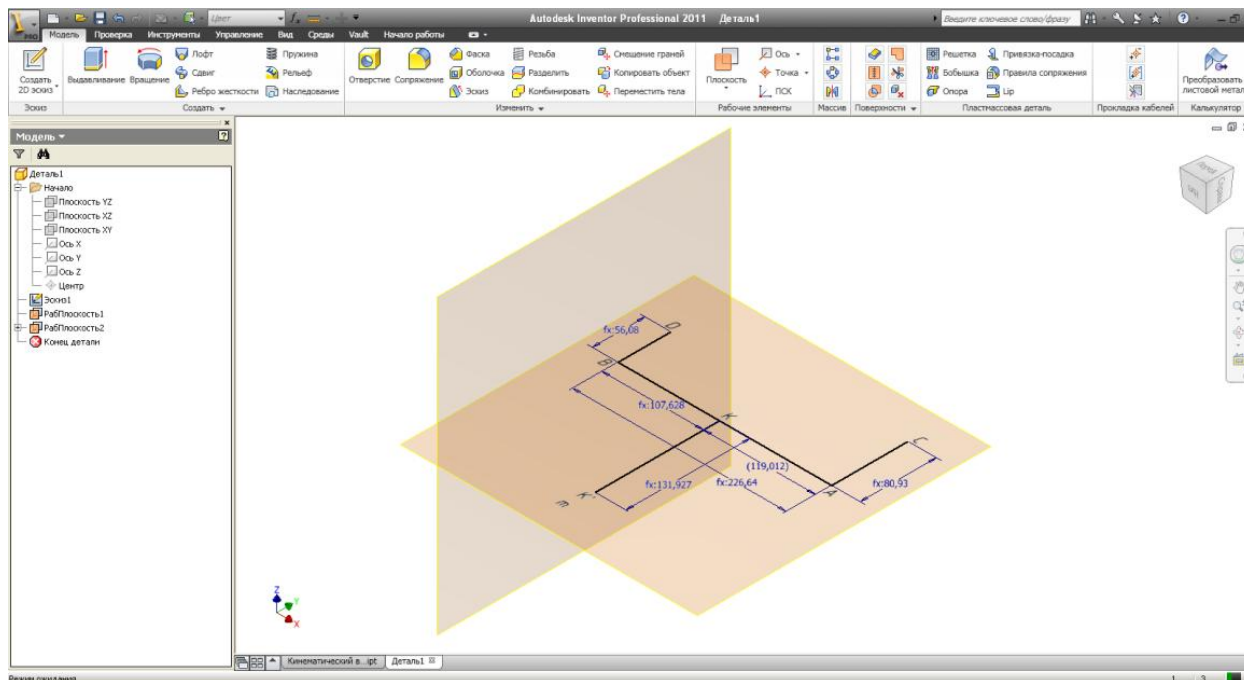


Рис. 5. «Робоча площина 2»

Перпендикулярно до «Робочої площини 1» створена «Робоча площина 3», що проходить через відрізок AC «Ескізи 1» (рис. 6).

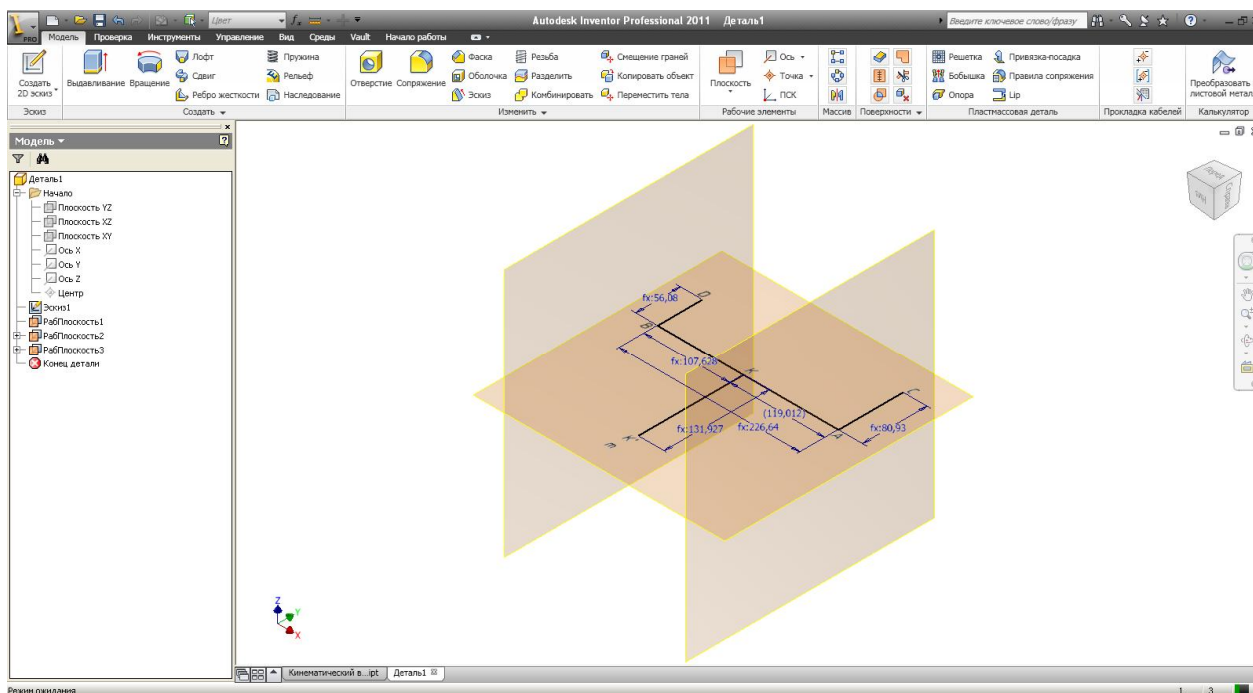


Рис.6 «Робоча площина 3»

Перпендикулярно до «Робочої площини 1», «Робочої площини 2» та «Робочої площини 3» створена «Робоча площина 4», що проходить через відрізок АВ «Ескізи 1» (рис. 7).

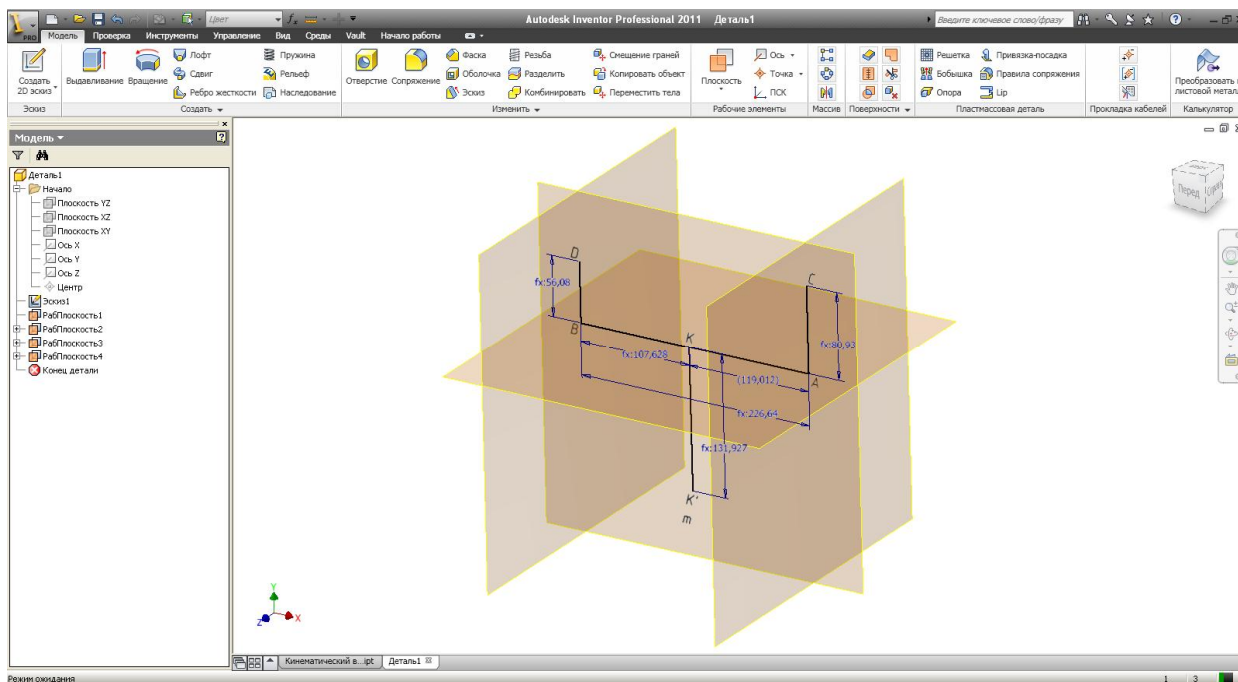


Рис.7. «Робоча площина 4», що проходить через відрізок АВ

На «площині 2» створений «Ескіз 2» (рис.8).

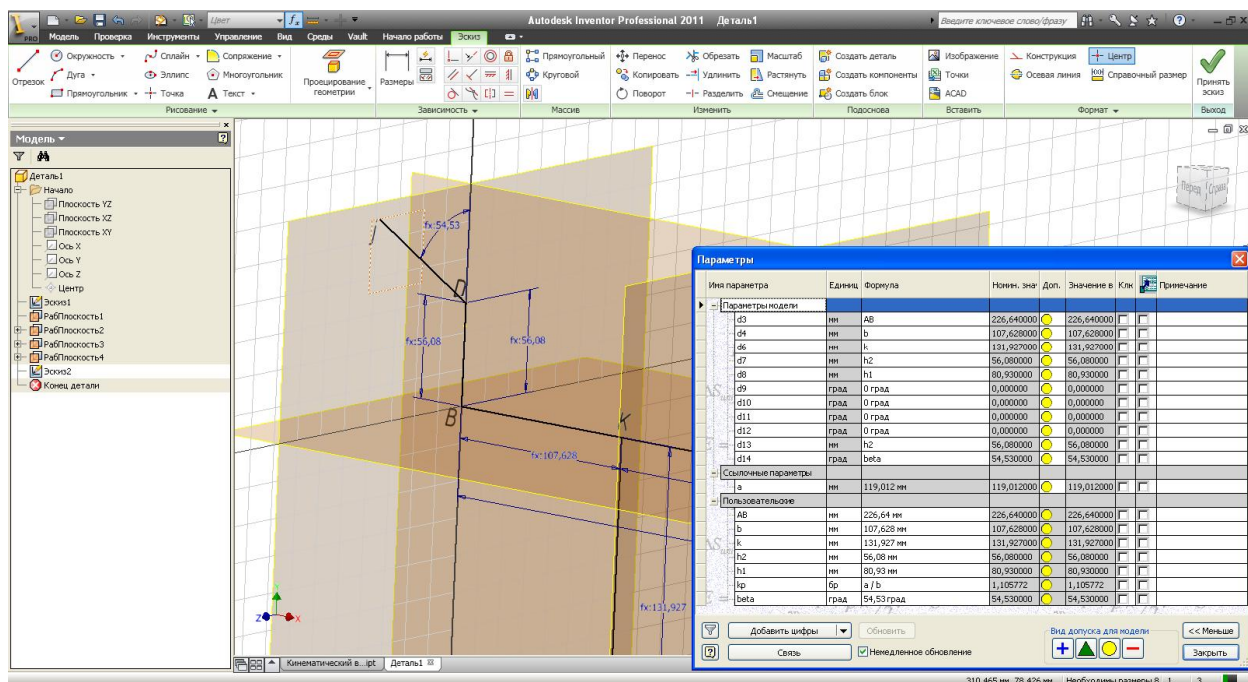


Рис.8. На «площине 2» створений «Ескіз 2»

Через відрізок Dj «Ескізи 2» створена «робоча вісь 1» (рис. 9).

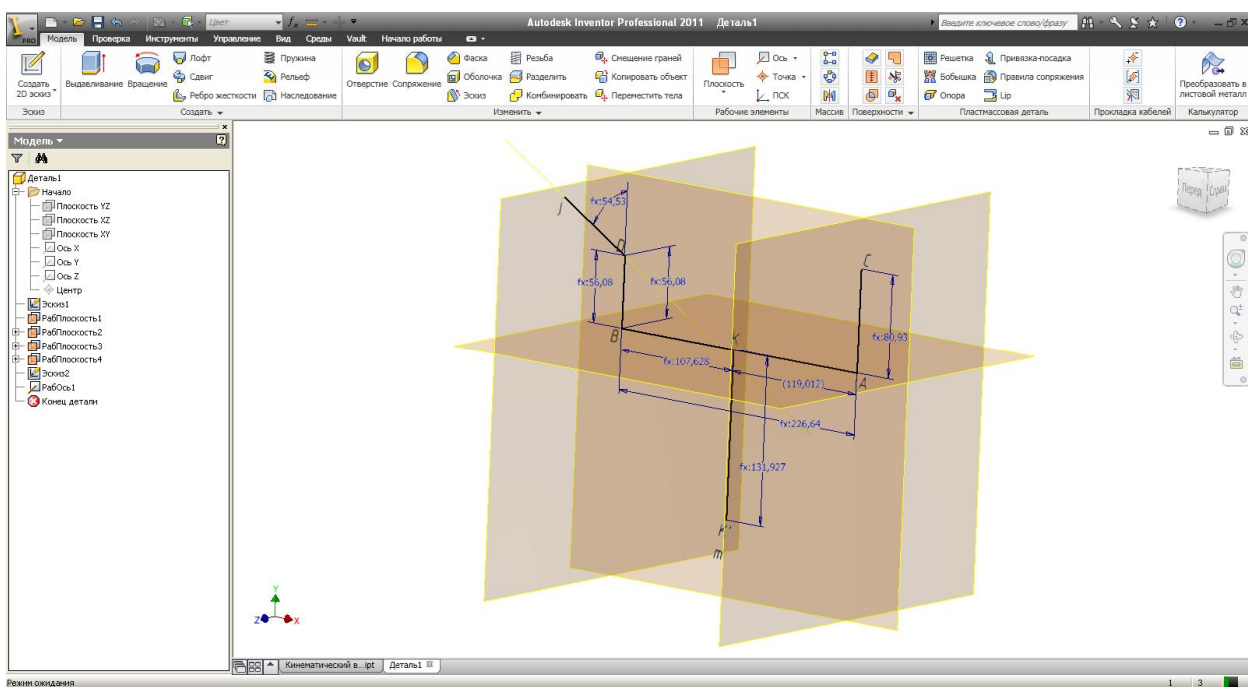


Рис. 9. «Робоча вісь 1»

На «Робочій площині 3» створений «Ескіз 3». Через відрізок C_i створена «робоча вісь 2» (рис. 10).

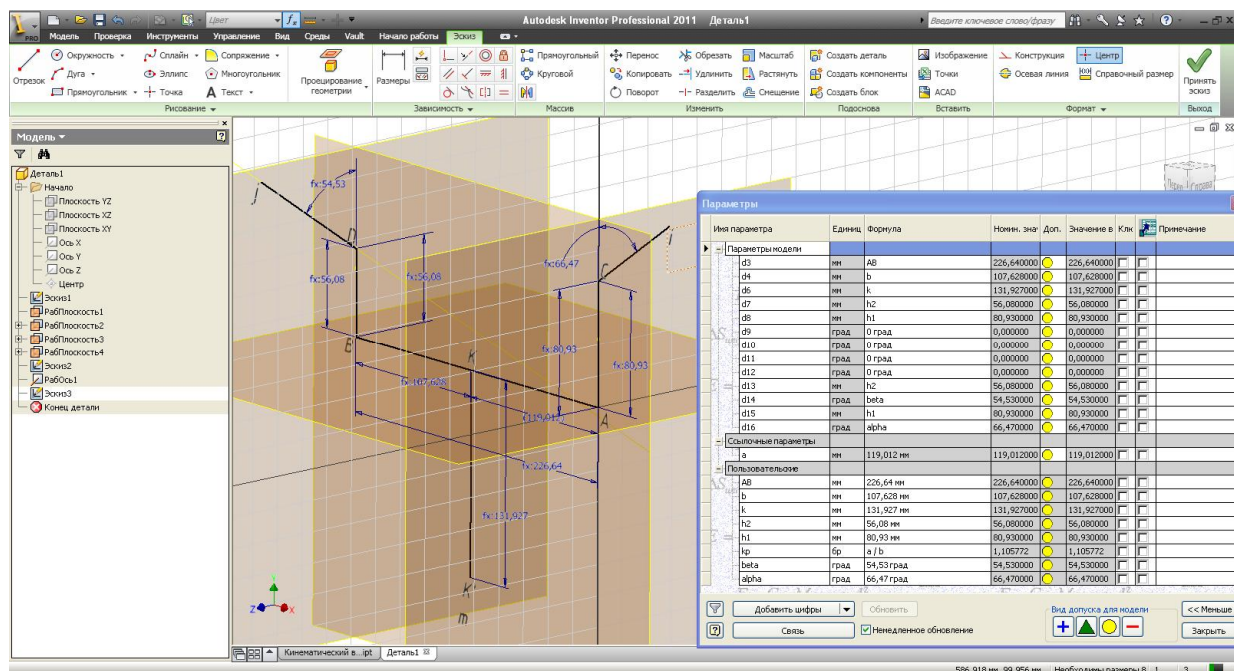


Рис.10. «Робоча вісь 2»

Через відрізок КК' «Ескізи 1» створена «робоча вісь 3», яка відповідає підсумковій осі m (рис. 11).

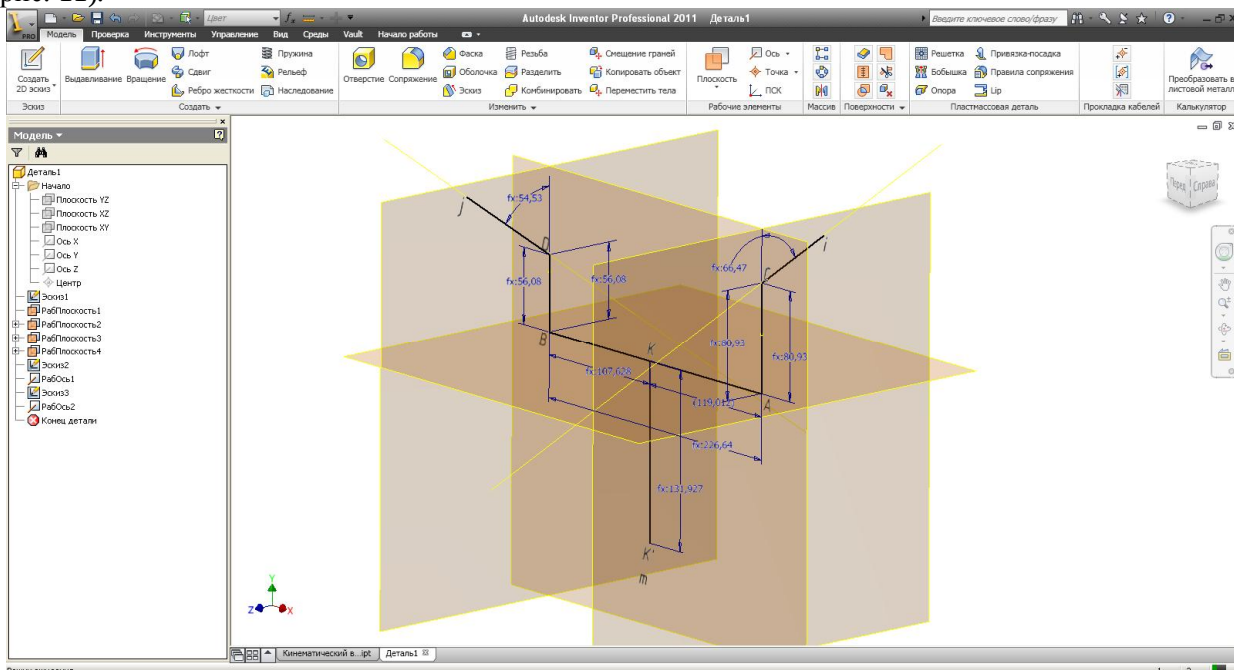


Рис.11. Підсумкова вісь m

В «Робочій площині 1» створений «Ескіз 4» (рис. 12). Коло проходить через точки D, C і K'.

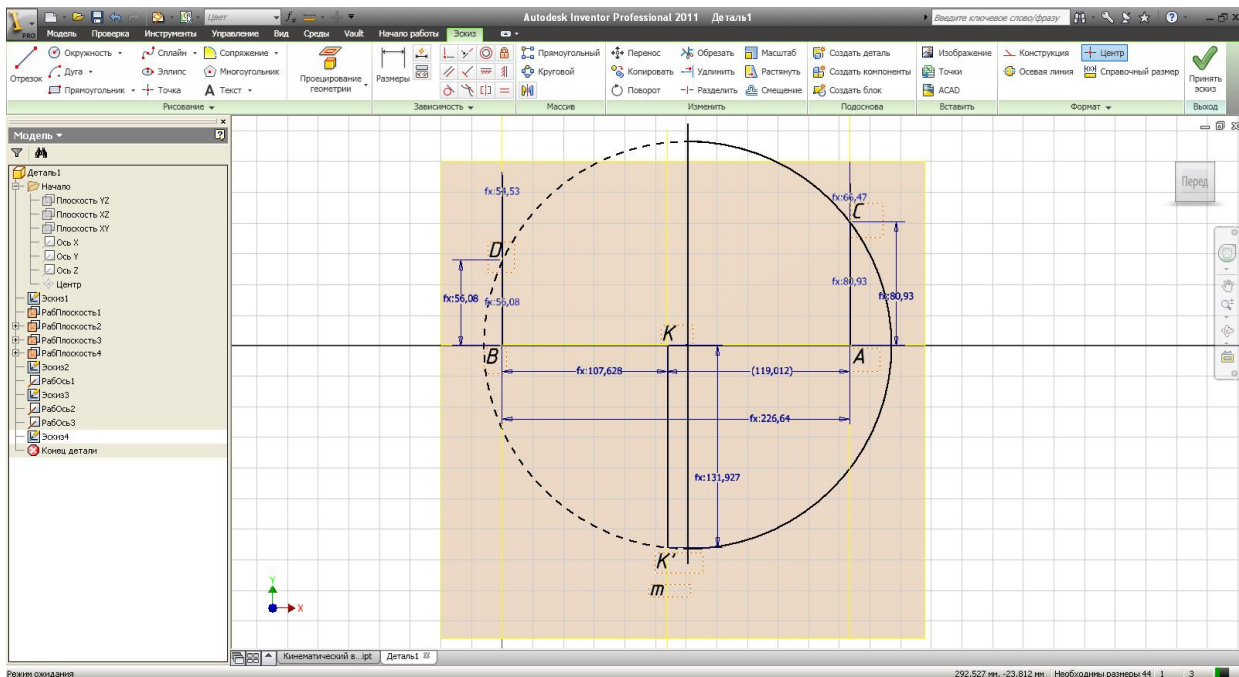


Рис.12. «Ескіз 4»

Виконано команду «Обертання» (рис. 13):

- ескіз – обрана дуга, позначена основною лінією;
- вісь – обраний відрізок, що проходить через центр дуги;
- результат - «Поверхня»;
- обмеження - «Повне коло».

Геометрична модель кінематичного гвинта виконана в САПР Autodesk Inventor, що є системою параметричного моделювання, тобто дозволяє довільно міняти розміри проєктованих аксодів, при цьому зберігаючи їх конфігурацію і цілісність.

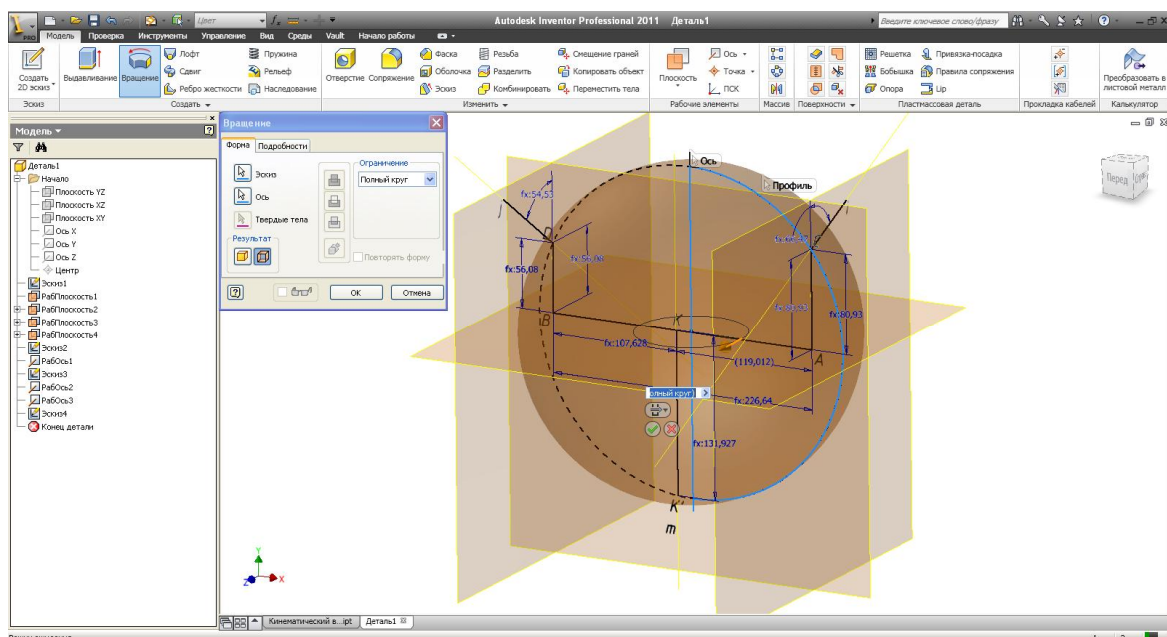


Рис.13. Отримана тривимірна модель просторового параметричного кінематичного гвинта

Зображенню просторового параметричного кінематичного гвинта відповідає сукупність послідовно виконуваних геометричних побудов.

Висновок. Геометричне моделювання вдосконаленого просторового параметричного кінематичного гвинта дозволяє визначити трудомісткі розрахунково-графічні операції побудови параметрів сімейства ліній контакту зачеплення, автоматично дозволяє значно підвищити точність профілювання зубчастих і гвинтових зачеплень в кінематичних парах, автоматизувати процес обкатки в машинобудуванні.

1. Подкорытов А.Н. Определение интерференции и профилирование сопряженных винтовых нелинейчатых поверхностей со сложным криволинейным профилем применительно к червячным фрезам. Прикладная геометрия и инженерная графика. Омск, 1972.
2. А.М. Подкоритов, Ісмаїлова Н.П. Основи формування поверхонь //Прикладна геометрія та інженерна графіка. Праці / Таврійська державна агротехнічна академія – Вип. 7, т.38. – Мелітополь: ТДАТА, 2008. – С.16-20.
3. Ісмаїлова Н.П.Твірдітільне моделювання спряжених поверхонь на базі параметричного кінематичного гвинта// "Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – Вип. 21. ЛНТУ–Луцьк, 2015. – С.15-20
4. Подкоритов А. М., Ісмаїлова Н. П. Теоретичні основи спряжених квазігвинтових поверхонь, що виключають інтерференцію [Текст]: монографія //Херсон : ФОП Грінь Д. С., 2016. – 330 с

УДК 62-231:621.9.04

Кириченко А. М. проф. д.т. н., Аль Ібрахімі Метак М.А. аспірант
Центральноукраїнський національний технічний університет

РОБОЧИЙ ПРОСТІР ТА ОСОБЛИВОСТІ 5-ОСЬОВОГО ПАРАЛЕЛЬНОГО ВЕРСТАТА З НАДЛИШКОВИМИ ПРИВОДАМИ

Кириченко А.М., Аль Ібрахімі Метак М.А. Робочий простір та особливості 5-осьового паралельного верстата з надлишковими приводами. Високошвидкісна обробка є одним з основоположних підходів у підвищенні продуктивності праці і якості одержуваних деталей в машинобудівному виробництві, яких можна досягти при використанні для обробки багатокомпонентних 5-осьових паралельних верстатів з надлишковими приводами. У процесі розроблення багатокомпонентних 5-осьових паралельних верстатів одним з головних недоліків є вкрай низький коефіцієнт використання робочого простору. Тому, актуальним стає дослідження робочого простору 5-осьових паралельних верстатів з надлишковими приводами для розробки більш ефективних і продуктивних маніпуляторів. Були розглянуті останні публікації у відкритому доступі щодо існуючих багатокомпонентних 5-осьових паралельних верстатів. Дослідження робочого простору і описання 5-осьового паралельного верстата з надлишковими приводами при обробці заготовок. У статті необхідно дослідити робочий простір та особливості розробленого 5-осьового паралельного верстата з надлишковими приводами. 5-осьові паралельні верстати з надлишковими приводами на основі паралельних кінематичних структур мають найбільший об'єм робочого простору. Положення виконавчого органа під час обробки поверхонь забезпечується зміною довжини кожної кінематичної ланки залежно від геометричної форми поверхні, що обумовлює відповідні траєкторії формоутворюючих рухів інструмента. У роботі досліджено робочий простір та особливості розробленого 5-осьового паралельного верстата з надлишковими приводами. Показано, що зменшення об'єму робочого простору відбувається за рахунок зменшення його діаметральних розмірів, а також висоти робочого простору.

Ключові слова: робочий простір; паралельний верстат; надлишковий привід, обробка.

Кириченко А. Н., Аль Ибрахими Метак М.А. Рабочее пространство и особенности 5-осевого параллельного станка с избыточными приводами. Высокоскоростная обработка является одним из основополагающих подходов в повышении производительности труда и качества получаемых деталей в машиностроительном производстве, которые достигаются с использованием для обработки многокомпонентных 5-осевых параллельных станков с избыточными приводами. В процессе разработки многокомпонентных 5-осевых параллельных станков одним из главных недостатков является крайне низкий коэффициент использования рабочего пространства. Поэтому, актуальным становится исследование рабочего пространства 5-осевых параллельных станков с избыточными приводами для разработки более эффективных и производительных манипуляторов. Были рассмотрены последние публикации в открытом доступе относительно существующих многокомпонентных 5-осевых параллельных станков. Исследование рабочего пространства и описание 5-осевого параллельного станка с избыточными приводами при обработке заготовок. В статье необходимо исследовать рабочее пространство и особенности разработанного 5-осевого параллельного станка с избыточными приводами.. 5-осевые параллельные станки с избыточными приводами на основе параллельных кинематических структур имеют наибольший объем рабочего пространства. Положение исполнительного органа во время обработки поверхностей обеспечивается изменением длины каждого кинематического звена в зависимости от геометрической формы поверхности и обуславливается соответствующими траекториями формообразующих движений инструмента. В работе исследовано рабочее пространство и особенности разработанного 5-осевого параллельного станка с избыточными приводами. Показано, что уменьшение объема рабочего пространства происходит за счет уменьшения его диаметральных размеров, а также высоты рабочего пространства.

Ключевые слова: рабочее пространство; параллельный станок; избыточный привод, обработка.

Kyrychenko Andriy, Al Ibrahimi Metak M.A. Workspace and singularities of 5-axis parallel robot with redundant actuation. High-speed machining is one of the fundamental approaches to increasing the productivity and quality of the parts produced in the machine-building industry, which are achieved with the use of multi-component 5-axis parallel robot with redundant actuation. In the process of developing multi-component 5-axis parallel robot one of the main disadvantages is the extremely low utilization rate of the workspace. Therefore, it is relevant to study the working space of 5-axis parallel robot with redundant actuation to develop more efficient and productive manipulators. The last publications were publicly available in relation to existing 5-axis parallel robot. Study of the workspace and singularities of a 5-axis parallel robot with redundant actuation when processing blanks. In the article it is necessary to investigate the workspace and singularities of the developed 5-axis parallel robot with redundant actuation. 5-axis parallel robot with redundant actuation based on parallel kinematic structures have the largest amount of workspace. The position of the executive body during the surface treatment is provided by changing the length of each kinematic link, depending on the geometric shape of the surface and is determined by the corresponding trajectories of the shaping movements of the tool. The workspace and singularities of the developed 5-axis parallel robot with redundant actuation are investigated. It is shown that the reduction in the volume of the working space is due to the reduction of its diametrical dimensions, as well as the height of the workspace.

Keywords: workspace; parallel robot; redundant actuation, processing.

Актуальність теми дослідження. Високошвидкісна обробка є одним з основоположних підходів у підвищенні продуктивності праці і якості одержуваних деталей в машинобудівному виробництві. Високошвидкісна обробка отримала розвиток в різних сферах промисловості [1]. Це,

в першу чергу, підвищує міцність, знижує вагу конструкції і підвищує надійність. Процес складання при переході до монолітних деталей істотно спрощується і значно знижується трудомісткість пригонювальних та слюсарно-складальних робіт. Зазначені тенденції вимагають створення металорізальних верстатів з новими функціональними можливостями [2, 3], що забезпечують високу швидкість знімання металу при чорновій обробці металів і сплавів, високу якість поверхні і точність обробки; при фінішних операціях відсутність суттєвих вібрацій при обробці (вібрації в процесі різання негативно впливають на довговічність і надійність роботи деталі в процесі експлуатації). Таких показників можна досягти, використовуючи для обробки багатокомпонентні 5-осьові паралельні верстати з надлишковими приводами. Однак і такі багатокомпонентні верстати потребують вдосконалення та розширення орієнтаційних можливостей.

Постановка проблеми. Особливості надлишкових механізмів обумовлюють значні відмінності кінематичних залежностей, методів розрахунку точності та жорсткості, потребують відповідного уточнення алгоритмів управління, виходу у початкове положення та калібрування верстата. Одним з головних недоліків обладнання, побудованого на основі паралельних структур, є вкрай низький коефіцієнт використання робочого простору [2-4]. У верстатах з традиційними послідовними кінематичними зв'язками всі лінійні переміщення вихідних органів приводу подач пропорційно переносяться на об'єм робочого простору. У механізмах паралельної структури ситуація набагато складніша. Кожна штанга такого механізму повинна забезпечити переміщення робочого органу по всім можливим координатам, при цьому уникнувши особливих положень. Тому, виникає необхідність у дослідженні робочого простору 5-осьових паралельних верстатів з надлишковими приводами.

Аналіз останніх досліджень і публікацій. Широкого застосування в машинобудуванні знайшли вертикальні 5-ти координатні верстати, що мають координати X, Y, Z, A, C. Типовим представником таких верстатів є мод. CFV5 фірми Cincinnati Machine Ltd. CFV5 – це 5-осьовий вертикальний обробний центр, який ідеально підходить як для п'ятисторонньої обробки, так і для виготовлення прес-форм і шаблонів [5, 6]. Верстат може функціонувати з різними шпинделями, системами видалення стружки і подачі охолоджуючої рідини, а також безліччю додаткових пристосувань, що дозволяє його легко адаптувати до різноманітних практичних задач. Вбудований глобусний стіл забезпечує можливість проводити одночасну 5-ти координатну обробку деталей завдяки своїй місткості і гнучкості переміщень. При максимальному навантаженні до 100 кг швидкість обертання осі C на CFV5 може досягати 12000 град / хв, а швидкість повороту осі A – 12000 град / хв.

Горизонтальні 5-ти координатні верстати (X, Y, Z, A, B) для реалізації кутових рухів за координатами A і B оснащуються шпиндельною головкою, що представляє собою накладені один на інший поворотні столи. Удосконалюючи цей принцип, фірма Cincinnati Machine Ltd. розробила шпиндельну головку, в якій для реалізації рухів кутових нахилів A, B застосовуються кульково-гвинтові передачі (ballscrews) [7]. Шпиндельна головка забезпечує нахил шпинделя за координатами A і B в межах $\pm 40^\circ$. Цю конструкцію головки шпинделя фірма Cincinnati Machine Ltd. широко застосовує в своїх 5-ти координатних верстатах, призначених для роботи в різних галузях промисловості.

Такий же принцип покладено в основу роботи головки шпинделя, якою оснащений 5-ти координатний верстат останньої розробки мод. Automax Horizontal High Speed Plate Mill [6]. Діапазон нахилу шпинделя за координатами A і B складає $\pm 45^\circ$. Шпиндель має частоту обертання до 24000 об / хв. Діапазон подач – до 40 000 мм / хв. Верстат дозволяє обробляти в монолітних деталях перегородки, товщиною до 0,5 мм. Альтернативним напрямком розвитку конструкцій шпиндельних головок 5-ти координатних верстатів для різних сфер промисловості є використання в верстатах механізмів паралельної кінематики.

Ряд зарубіжних фірм направив свої зусилля на створення 5-ти координатних верстатів по структурі P2C3K5 з використанням триподів. Фірма DS Technologie досягла добрих результатів у створенні 5-ти координатних верстатів для високошвидкісної обробки деталей з твердого алюмінію [5]. Свої верстати вона побудувала на базі Z3 модуля з несучим шпинделем. Модуль забезпечує шпинделю 3 переміщення: одне лінійне 670 мм і два кутових $\pm 40^\circ$ (нахили шпинделя в двох взаємно перпендикулярних площинах). Конструкція Z3 модуля побудована на базі механізму паралельної кінематики (трипода). Шпиндель має потужність 80 кВт і максимальну частоту обертання 30000 об / хв, конус шпинделя HSK-A63 / 80.

Верстат мод. ECO SPEED, що призначений для високошвидкісної обробки великогабаритних деталей з алюмінієвих сплавів з суцільних заготовок (плит) середнього розміру, має високу точність обробки (розмір, форма, розташування) і високу якість поверхні, що виключають наступні слюсарно-пригоночні і полірувальні роботи до і в процесі складання виробів [6].

Обробний центр Verne від Fatronik являє собою 5-осеву гібридну паралельну кінематичну машину, шпиндельна головка якої знаходиться на паралельному модулі DOF, а відрядна частина розміщена на поворотному столі, який має підмножину зсувів, розмірності 2, які створюються добутком двох незалежних обертань навколо осі A і C [7].

У роботах [4, 8-10] розглянуто і запропоновано 5-осьовий паралельний верстат з надлишковими приводами. Розроблений багатокоординатний верстат на основі надлишкового механізму паралельної структури має 5 ступенів вільності 2R3T (2 обертальних і 3 поступальних ступені вільності) з розширеними можливостями орієнтації, що досягається виключенням особливих положень за рахунок використання надлишкових приводів і спеціальної конструкції робочого органа, для використання у верстатах паралельної структури. Для використання принципу надлишковості приводів у багатокоординатному обладнанні розроблено схему верстата паралельної структури з 5 керованими координатами і 6 приводами штанг змінної довжини, який має спеціальний робочий орган з спільною віссю шарнірних опор. Робочий орган верстата з'єднується з основою шістьма ланками змінної довжини, причому шарніри робочого органа мають спільну вісь, яка співпадає з віссю шпинделя верстата. Використання спеціального робочого органа дозволяє забезпечити надлишковість механізму паралельної структури вже при кількості приводів 6, тоді як для надлишкової платформи Гауфа-Стюарта необхідно хоча б 7 приводних ланок. Привід переміщення вихідного органа верстата з шістьма штангами на основі паралельних кінематичних структур містить станину, механізми подачі шести штанг, які пов'язані з вихідним органом. Вихідний орган складається з двох частин, поєднаних між собою шарніром з трьома степенями вільності. При цьому три штанги приєднуються до однієї частини вихідного органа і мають конструкцію, що забезпечує переміщення цієї частини вихідного органа по трьох лінійних координатах без зміни кутової орієнтації відносно станини, а інші три штанги приєднуються до іншої частини вихідного органа.

Виділення недосліджених раніше частин загальної проблеми. Розроблений 5-осьовий паралельний верстат з надлишковими приводами не підлягав описанню особливостей та дослідженню робочого простору при обробці заготовок.

Постановка завдання (цілей статті). У статті необхідно дослідити робочий простір та особливості розробленого 5-осьового паралельного верстата з надлишковими приводами.

Виклад основного матеріалу. Можливість реалізації схеми формоутворення заданої поверхні на верстаті залежить від відносної орієнтації траєкторій виконавчих рухів, яка визначається компонованням обладнання та наявністю допоміжних рухів, що дозволяють змінювати взаємне розташування виконавчих органів.

Робочий простір у верстатах з традиційними послідовними кінематичними зв'язками буде зменшуватися за рахунок відстані між вершиною інструмента та віссю обертання вихідної ланки. У механізмах паралельної структури всі переміщення повинні забезпечуватися за рахунок переміщення повзунів штанг або зміни їх довжини. Таким чином постає питання, що ж розглядати у якості об'єму робочого простору. Щодо верстатів з традиційною кінематикою, то під робочим простором розуміється просторова сукупність всіх можливих положень інструмента. У механізмах паралельної структури виділяють декілька зон можливого розташування інструмента [3, 9, 10]. Для коректного порівняння даних характеристик між верстатами з традиційною кінематикою та з паралельною кінематикою, очевидно щодо останніх, необхідно використовувати робочий простір (а відповідно і його розміри) з повною заданою рухомістю вихідного органа. Для таких верстатів це складна задача, оскільки робочий простір повинен вмістити і сам робочий орган з інструментом, і заготовку. За розрахованих значень коефіцієнта використання робочого простору навіть незначне зростання об'єму робочого простору призводитиме до помітного збільшення габаритів верстата в цілому. Звичайно, можна піти екстенсивним шляхом і збільшити довжину штанги. В такому випадку коефіцієнт використання робочого простору виросте, однак зростуть і габарити верстата.

Часткове вирішення цієї задачі можна знайти у використанні комбінації механізмів для надання руху подачі як інструменту, так і заготовці [11]. Практично будь-який вузол, що надає

один елементарний рух (як лінійний, так і кутовий), можна розглядати як механізм паралельної структури з переміщенням по одній координаті. Таке порівняння коректне, оскільки той же повзун у механізмах зі штангами постійної довжини є саме таким вузлом.

Припустимо, що проблема неефективності забезпечення необхідних розмірів робочого простору верстатів з паралельними кінематичними структурами з повною заданою рухомістю вихідного органу приводу подачі напряму залежить від кількості штанг, що приводять його у рух. Відомо, що для забезпечення переміщень по n координатам у механізмі паралельної структури повинно бути як мінімум стільки ж рухомих штанг. Якщо ж зменшити кількість штанг, то задача забезпечення необхідних розмірів робочого простору для верстатів з паралельними кінематичними структурами спроститься.

Відсутні при цьому координати переміщення інструменту пропонується надати заготовці. Розглянемо випадок шестикоординатного механізму паралельної структури, з якого будемо виключати ту чи іншу координату переміщення інструменту та надавати її заготовці. Довільно обираємо початкове положення інструменту вертикально зверху над закріпленою на столі з горизонтальною площиною заготовкою. У випадку нерухомої заготовки інструмент повинен мати доступ до всіх поверхонь заготовки, окрім площини кріплення. Відтак його лінійні переміщення по трьом координатам повинні перекривати розміри заготовки, а повороти навколо осей X та Y відносно початкового положення інструменту повинні становити 180° з підходами до відповідних площин обробки. У випадку використання обертового осьового інструменту для обробки різанням поворот робочого органу навколо осі інструмента є скоріше «певним доповненням», аніж технологічною необхідністю.

Розглянемо чотири рухи, які можна надати вихідному органу, на якому закріплена заготовка. Це поворот навколо осі інструмента та перпендикулярної до неї осі, прямолінійні рухи вздовж даних осей. Відсутні у розгляді два рухи кінематично-подібні випадку прямолінійного переміщення та повороту навколо перпендикулярної до осі інструмента осі.

Перший випадок – заготовка може обертатися навколо осі прийнятого початкового положення інструменту. Це призводить до суттєвого скорочення можливої кутової орієнтації інструмента навколо заданої точки робочого простору. Так, у випадку нерухомої заготовки інструмент повинен знаходитись у півсфері навколо даної точки. У випадку обертання заготовки, інструмент повинен знаходитись лише у $1/8$ сфери навколо заданої точки робочого простору. Інші поверхні заготовки, необхідні для обробки, будуть надані у зоні обробки шляхом повороту.

Другий випадок – заготовка може обертатися навколо осі, перпендикулярної до прийнятого початкового положення інструменту. Це також призводить до скорочення можливої кутової орієнтації інструмента навколо заданої точки робочого простору, однак удвічі меншої від попереднього випадку.

Третій випадок – заготовка здійснює рух подачі вздовж осі прийнятого початкового положення інструменту. В такому випадку лінійні переміщення інструменту можуть відбуватися лише у площині, але оскільки за поставленими умовами необхідно забезпечувати поворот навколо осей системи координат, переміщення всіх шарнірів робочого органу по одній чи рівнобіжним площинам не відбувається, і спрощення кінематики руху мінімальне. Вагомим зиском залишається обробка високих заготовок.

Четвертий випадок – заготовка здійснює рух подачі вздовж осі, перпендикулярної до прийнятого початкового положення інструменту. Цей випадок подібний до попереднього, з певною відмінністю у кінематиці. Ефект полягає у можливості обробки довгих заготовок.

Зміна умов роботи 5-осьового верстата з комбінованим механізмом паралельної структури при наданні заготовці переміщення вздовж однієї з координат при відповідному вилученні такого руху у робочого органу з інструментом показує, що найбільш доцільною з точки зору спрощення кінематики є схема з обертанням заготовки у горизонтальній площині навколо початкового вертикального розташування інструменту.

Надання заготовці лінійних переміщень дозволяє збільшити робочий простір, що підвищує ефективність конструкції верстата при обробці довгих заготовок.

5-осьові паралельні верстати з надлишковими приводами на основі паралельних кінематичних структур мають найбільший об'єм робочого простору. Зменшення об'єму робочого простору відбувається за рахунок зменшення його діаметральних розмірів (D), а також висоти (H) робочого простору (рис. 1).

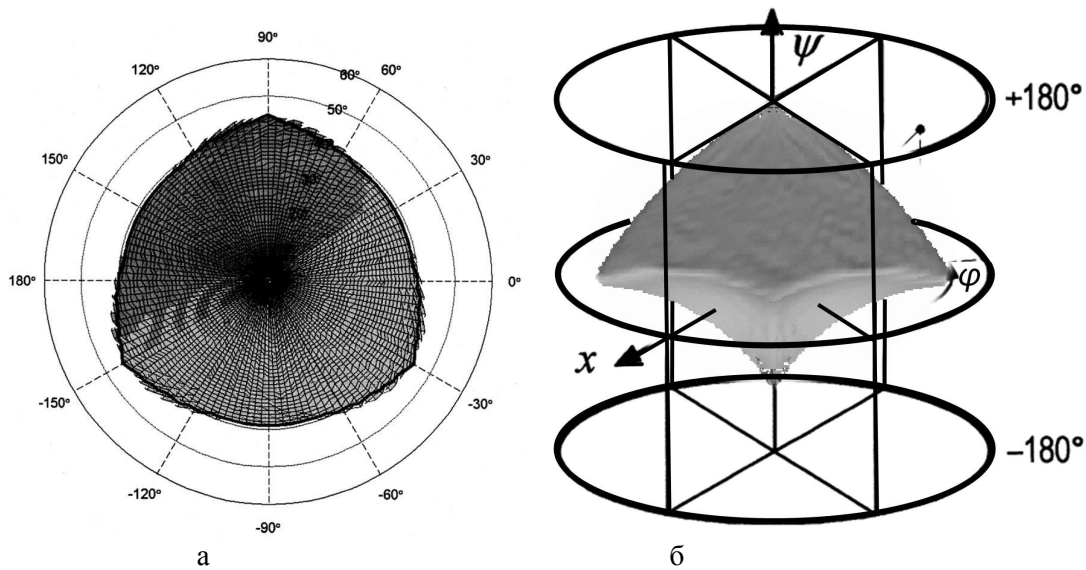


Рисунок 1. Робочий простір 5-осьового паралельного робота з надлишковими приводами на основі паралельних кінематичних структур: а – вид зверху; б – головний вид

Найбільший діаметр при найменшій висоті має 5-осьовий паралельний робот з надлишковими приводами на основі паралельних кінематичних структур. Поточне положення виконавчого органа під час обробки поверхонь забезпечується зміною довжини кожної кінематичної ланки залежно від геометричної форми поверхні, що обумовлює відповідні траєкторії формоутворюючих рухів інструмента.

Висновки відповідно до статті. У роботі досліджено робочий простір та особливості розробленого 5-осьового паралельного верстата з надлишковими приводами. Показано, що зменшення об'єму робочого простору відбувається за рахунок зменшення його діаметральних розмірів, а також висоти робочого простору. Положення виконавчого органа 5-осьового паралельного верстата з надлишковими приводами під час обробки поверхонь забезпечується зміною довжини кожної кінематичної ланки залежно від геометричної форми поверхні, що обумовлює відповідні траєкторії формоутворюючих рухів інструмента.

1. Aref, M. M. Geometrical workspace analysis of a cable-driven redundant parallel manipulator: KNTU CDRPM / M. M. Aref, H. D. Taghirad // *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice. - 2008. - P. 1958-1963.
2. Borras, J. Architectural singularities of a class of pentapods / Borras, J., Thomas, F., Torras, C. // *Mechanism and Machine Theory*, Issue 46 (8), 2011. - P. 1107-1120.
3. Cao, Y. Singularity Elimination of Stewart Parallel Manipulator Based on Redundant Actuation / Y. Cao, H. Zhou, B. Li, S. Long, M. Liu // *Advanced Materials Research*. - 2011. - Vols. 143-144. - P. 308-312.
4. Кириченко А.М., Аль Ібрахімі Метак. Верстат паралельної структури з надлишковими приводами та спеціальним робочим органом // *Комплексне забезпечення якості технологічних процесів та систем*. - 2017. - С. 136-137.
5. Müller, A. On the terminology and geometric aspects of redundant parallel manipulators // *Robotica*. - 2013. - Vol. 31(1). - P. 137-147.
6. Sartori, Natal G. Dual-space adaptive control of redundantly actuated parallel manipulators for extremely fast operations with load changes / G. Sartori Natal, A. Chemori, F. Pierrot // *Proceedings of IEEE International Conference on Robotics and Automation, IEEE*, Saint Paul, MN, USA, pp. 253-258, 2012.
7. Sirotkin, R., Serkov N. Experimental Researches of Accuracy, Rigidity and dynamic Properties – 5-axis Machining Centre Hexamech-1 *The 5th Chemnitz Parallel Kinematic Seminar* 2006, April 25-26. Conference Proceedings. Zwickau: Verlag Wissenschaftliche Scripten, 2006, pp.
8. Кириченко А. М., Гречка А. І., Аль-Ібрахімі Метак М. А. Визначення ефективної схеми верстата комбінованої структури // *Сучасні технології промислового комплексу*. - ХНТУ, Херсон. - 2016. - С. 57-60.
9. Кириченко А.М., Аль Ібрахімі Метак. Багатокоординатний верстат з надлишковим механізмом паралельної структури та спеціальним робочим органом // *Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація*, 2017, вип. 30. - С. 57-64.
10. Кириченко, А. М. Кінематика п'ятикоординатного верстата з паралельною структурою [Текст] / А. М. Кириченко, О. В. Шелепко, С. П. Салон // *Вісник ЧДГУ. Серія «Технічні науки»*. - Чернівці : ЧДТУ, 2013. - № 3 (67). - С. 100-104.
11. Luces, M. A. Review of Redundant Parallel Kinematic Mechanisms / M. Luces, J. K. Mills, B. Benhabib // *Journal of Intelligent and Robotic Systems*. - May 2017. - Vol. 86, Issue 2. - P. 175-198.

УДК 378.146

Кізим С. О., Ройко О. О., Юхта О. А.

Волинський коледж Національного університету харчових технологій

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ В САМОСТІЙНІЙ РОБОТІ СТУДЕНТІВ

Кізим С. О., Ройко О. О., Юхта О. А. Інформаційно-комунікаційні технології в самостійній роботі студентів. Сучасна освіта має створювати умови, за якими у студентів формується досвід самостійної діяльності, в результаті якої найбільш повно розкриваються індивідуальні здібності й створюються передумови для творчого розвитку майбутнього фахівця. Важливу роль в організації самостійної роботи студентів відіграє використання сучасних технічних засобів, інформаційно-комунікаційних технологій. В статті розглянуто можливість використання інформаційно-комунікаційних технологій в організації самостійної роботи студентів.

Ключові слова: самостійна робота, інформаційно-комунікаційні технології, «хмарне сховище», «хмарні сервіси»

Кизым С. А., Ройко А. А., Юхта А. А. Информационно-коммуникационные технологии в самостоятельной работе студентов. Современное образование должно создавать условия, по которым у студентов формируется опыт самостоятельной деятельности, в результате которой наиболее полно раскрываются индивидуальные способности и создаются предпосылки для творческого развития будущего специалиста. Важную роль в организации самостоятельной работы студентов играет использование современных технических средств, информационно-коммуникационных технологий. В статье рассмотрена возможность использования информационно-коммуникационных технологий в организации самостоятельной работы студентов.

Ключевые слова: самостоятельная работа, информационно-коммуникационные технологии, «облачное хранилище», «облачные сервисы»

Kizym S.O., Roiko O.O., Yukhta O. A. Information and communication technologies in independent work of students. Modern education should create conditions under which students develop experience of independent activity, as a result of which the most complete disclosure of individual abilities and the preconditions for the creative development of the future specialist. An important role in the organization of independent work of students is played by the use of modern technical means, information and communication technologies. The article considers the possibility of using information and communication technologies in the organization of independent work of students.

Keywords: independent work, information and communication technologies, "cloud storage", "cloud services"

Постановка проблеми. Інтеграція України в європейський освітній простір ставить перед вітчизняною вищою школою завдання формування особистості, яка здатна творчо мислити, швидко набувати нові знання та вміти їх застосовувати до розв'язання нових нестандартних ситуацій. Сучасний випускник ВНЗ має бути конкурентоспроможним на ринку праці, для чого у процесі навчання він повинен здобувати не лише вузькоспеціалізовані, але й системні фундаментальні знання, що сприяють цілісному сприйняттю наукової картини світу, інтелектуальному розвитку особистості та її адаптації до швидко змінних соціально-економічних умов і розвитку технологій. Перехід від знанневої парадигми підготовки фахівців «навчання на все життя» до компетентнісної «навчання протягом всього життя» також вимагає посилення фундаментальної підготовки.

Кредитно-модульна система організації навчального процесу передбачає підвищення ролі самостійної роботи студентів. Формування здатності до самостійного оволодіння новими знаннями, спроможності аналізувати отриману інформацію, розвитку творчого мислення стають першочерговими завданнями вищої школи у підготовці висококваліфікованих спеціалістів. У зв'язку з посиленням ролі самостійної роботи в навчально-пізнавальному процесі актуальною стає проблема її раціональної організації.

Аналіз досліджень. Проблема організації самостійної роботи студентів існувала завжди – вчені та педагоги-практики завжди приділяли багато уваги вивченню різних аспектів, пов'язаних з самостійною роботою. Проблеми організації самостійної роботи студентів досліджували М.Г. Гарунов, Е.В. Гапон, В.А. Козаков, Л.І. Лутченко, З.І. Слєпкань, В.О. Швець та ін. Управління самостійною роботою студентів у позааудиторний час займалися Л.В. Клименко, В.П. Шпак та ін. Навчання студентів умінню планувати свою пізнавальну діяльність досліджували О.М. Козак, Н.П. Красницький та ін. Системний підхід в організації самостійної роботи студентів досліджувався в роботах Г.М. Гнітецької, Є.Г. Фомкіної, Н.В. Ванжі та ін. Незважаючи на значну кількість досліджень, присвячених самостійній роботі, проблема організації самостійної роботи студентів в сучасних умовах з використанням інформаційно-комунікативних технологій висвітлена недостатньо.

Мета статті. Проаналізувати використання інформаційно-комунікаційних технологій в організації самостійної роботи студентів у системі підготовки фахівців технічних спеціальностей, що дозволить сприяти формуванню самостійності студентів, засвоєнню знань, формуванню вмінь та навичок.

Виклад основного матеріалу: Організація навчання студентів технічних спеціальностей з нерівномірним рівнем підготовки викликає на практиці значну кількість проблем педагогічного, методичного та психологічного характеру: не всі студенти – учорашні школярі – звикли постійно працювати самостійно та виконувати домашні завдання; якщо домашні завдання не є індивідуальними, починає процвітати глобальне списування; якщо задавати індивідуальне домашнє завдання і його не перевіряти (за браком часу на перевірку), це приведе до того, що практично ніхто не буде його виконувати; якщо усі завдання перевіряти, виникають, крім браку часу, також окремі технічні проблеми. У таких умовах на перший план виходять проблеми підвищення результативності самостійної (індивідуальної) роботи студентів, особливо першокурсників, розробки дидактично обґрунтованих підходів до її проектування та організації, впровадження ефективних методів управління її виконанням. Необхідно виробити рекомендації до організації самостійної роботи, яка б сприяла отриманню студентами міцних знань, а також створювала б умови для плідної роботи викладацького складу. Останнім часом особлива увага суспільства та професійного педагогічного співтовариства звернена до різних аспектів модернізації сучасної освіти. Домінує думка, що якість освіти на всіх її ступенях останнім часом значно погіршала, багато абітурієнтів приходять до ВНЗ, не маючи конкретних знань з багатьох питань. Тому адаптація у ВНЗ першокурсників і організація самостійної роботи, впровадження оптимального співвідношення загальних, групових та індивідуальних форм організації навчального процесу у ВНЗ, раціонального застосування сучасних методів та засобів навчання на різних етапах підготовки фахівців є надзвичайно актуальною.

При вивченні фундаментальних дисциплін велике значення має засвоєння базових понять, розуміння їх властивостей, усвідомлення взаємозв'язку з викладенням наступного матеріалу, аналіз та вивчення доведень тверджень та теорем. Левова частка успішного засвоєння матеріалу залежить від систематичної самостійної роботи студента. На лекціях теоретичний матеріал викладається у поглибленій формі з доведеннями теорем та їхніх наслідків, а на практичних заняттях відпрацьовується розуміння основних понять, умінь та навички розв'язування прикладів на основі теорії. При зменшенні значної кількості аудиторних годин та відповідно збільшенні годин для самостійної роботи детальне вивчення значної кількості теорем та їх доведень, а також систематизація знань в основному залишаються на самостійне опрацювання студента. Теоретична частина самостійної роботи контролюється на колоквіумах. Досвід свідчить, що більше половини студентів на момент здачі колоквіуму не опрацьовують доведення теорем, тому колоквіум вважається першим підготовчим кроком в опануванні теорії. Контроль практичної частини навчального курсу переважно відбувається у формі аудиторних контрольних робіт та захисту індивідуальних домашніх завдань). Відповідно до сучасних умов навчання самостійна робота студентів повинна здійснюватись як при вивченні нового матеріалу, так і в процесі формування умінь та навичок. Отже, потрібно так організувати навчання студентів, щоб їх самостійна діяльність реалізувалася при всіх видах роботи: на лекції при написанні конспекту, при опрацюванні літератури, на практичних, семінарських та лабораторних заняттях, при виконанні домашнього завдання, контрольних та курсових робіт, науково-дослідної роботи та ін. Поліпшенню результатів сприяє органічне поєднання традиційних і новітніх засобів навчання, поступове впровадження у навчальний процес програмних засобів навчального призначення, раціональне поєднання традиційних методів та засобів навчання з сучасними інформаційними технологіями.

Особливої уваги потребують мотиваційне та технологічне забезпечення самостійної аудиторної та позааудиторної пізнавальної діяльності студентів.

Інформаційно-комунікаційні технології, швидке їх вдосконалення та використання в усіх сферах діяльності людини надають практично необмежені можливості розміщення, збереження, обробки та доставки інформації будь-якого об'єму, змісту на необмежені відстані. В цих умовах на перший план у підготовці фахівців висувається цілеспрямована робота із самоосвіти.

Співвідношення часу, який відводиться на аудиторну і самостійну роботу становить 1/3,5. Вимоги до самостійної роботи в Україні регламентує «Положення про організацію навчального процесу ВНЗ». Зокрема встановлено, що навчальний час, відведений для самостійної роботи

повинен становити не менше 1/3 загального обсягу та не більше 2/3 загального обсягу навчального часу студентів, що передбачено для вивчення конкретної дисципліни. Таке співвідношення базується на великому дидактичному потенціалі цього виду навчальної діяльності студентів, адже, знання, отримані студентами самостійно, завдяки власному досвіду, думці і дії, будуть насправді міцні. Навчальний матеріал, що опрацьовується власноруч, коли самостійно (індивідуально) виконується завдання від його постановки до аналізу отриманих результатів, засвоюється значно краще, ніж прослуханий під час лекції.

Самостійна робота студентів в даний час передбачає ефективне використання інформаційно-комунікаційних технологій, тобто засобів навчання нового покоління, а саме електронних засобів навчального призначення: електронні навчальні посібники, прикладні програмні засоби, мультимедійні курси, бібліотека електронних наочностей, електронні атласи, це використання комп'ютера (графічний і текстовий редактори, робота у комп'ютерних мережах), використання інформаційних джерел Інтернету.

Самостійна робота і творча діяльність студентів на сучасному етапі розвитку вищої освіти стають не лише засобом, а й метою навчального процесу. Все більшої ваги набирає необхідність організації та контролю цієї діяльності, методичного забезпечення, а також визначення ролі викладача в цьому процесі. Педагоги постійно шукають ефективні види організаційно-методичної діяльності, які б сприяли досягненню мети у підготовці фахівців, оскільки самостійна робота забезпечує можливість сформувати необхідний рівень і об'єм компетенцій, знань, навичок; виробити психологічну установку на самостійне поповнення своїх знань.

Впровадження в освітній процес інформаційно-комунікаційних технологій супроводжується збільшенням обсягів самостійної роботи студентів, що потребує постійної підтримки навчального процесу з боку викладачів. Важлива роль належить консультаціям, які ускладнюються з погляду дидактичних цілей: вони зберігаються як самостійні форми організації навчального процесу і водночас є елементами інших форм навчальної діяльності (лекції, практики, семінари, лабораторні практикуми).

Ефективність використання засобів інформаційних технологій у навчальному процесі залежить від успішності розв'язання завдань методичного характеру, пов'язаних з інформаційним змістом і способом використання автоматизованих систем навчання. Тому автоматизовані системи навчання доцільно розглядати як програмно-методичні комплекси (сукупність програмно-технічних засобів і реалізованих з їхнім використанням методів (методик) навчання, призначених для розв'язання конкретних завдань навчального процесу).

Застосування у самостійній роботі студентів інформаційно-комунікаційних технологій має низку суттєвих переваг у порівнянні з традиційними технологіями:

- забезпечує оптимальну для кожного конкретного студента послідовність, швидкість сприйняття матеріалу, розбору прикладів;
- формує навички аналітичної і дослідницької діяльності;
- створює можливості самоконтролю якості здобутих знань, умінь, навичок та компетенцій;
- значно заощаджує час студента, потрібний для вивчення відповідного курсу.

Використання інформаційно-комунікаційних технологій у педагогічному процесі впливає на характер навчально-пізнавальної діяльності студентів, активізує самостійну роботу студентів з різними електронними засобами навчального призначення. Найефективнішим є застосування інформаційно-комунікаційних технологій для відпрацювання навичок і умінь, необхідних для професійної підготовки.

Доступність технологій Веб розповсюджується на освітній процес. Такі технології надають можливість студентам у виборі індивідуального виду навчання, дослідження. Вони передбачають спільні способи роботи та гарантії збереження авторських прав. Перевагою технологій Веб є можливість індивідуалізації навчального процесу шляхом складання завдань та розширення діапазону знань зі спеціальності та в середовищі Інтернет. Технології Веб – це набір Інтернет-послуг та тренувань, що надають рівне право голосу кожному користувачеві. Такі сервіси дозволяють брати участь у різноманітних співтовариствах з метою одержання та розповсюдження досвіду. Розповсюдження знань у середовищі Інтернет залежить від функціонального використання браузеру, який забезпечує особисту безпеку в процесі роботи в Інтернет, постійно здійснюючи відправлення й одержання інформації, її обробку.

Таким чином, технології Веб – це певні етапи становлення та розвитку інформаційно-комунікаційних технологій та відповідного інформаційно-комунікаційного простору глобальної мережі Інтернет.

Впровадження в освітній процес Веб-технологій сприяє якісній самостійній роботі, особливо в процесі впровадження кредитно-модульної системи, здійсненню досліджень, а також закріпленню навчального матеріалу, розширенню та поглибленню знань. Все це сприяє індивідуалізації самостійної роботи студента, робить її особисто орієнтованою.

Сьогодні он-лайн сервіси на базі технологій хмарних обчислювань пропонують послуги, для користування якими потрібні лише пристрій з встановленим браузером та доступ до Інтернету. А, оскільки, майже кожен студент має такий електронний пристрій (notebook, netbook, планшет, телефон), то цього вже достатньо для роботи з хмарними технологіями.

В загальному випадку, сервіси хмарних обчислень це он-лайн додатки, доступ до яких забезпечується за допомогою будь-якого інтернет-браузера. Користувачеві не потрібно мати потужний комп'ютер для запуску специфічного програмного додатку, достатньо звернутися через Інтернет до провайдера, сплатити послугу або, в ідеальному випадку – отримати її безкоштовно.

Отже, впровадження хмарних технологій в освітній процес і використання їх для організації самостійної роботи студентів забезпечує:

- ефективне використання аудиторного фонду навчального закладу (немає необхідності створювати спеціально оснащені комп'ютерні класи);
- скорочення фінансових затрат на підтримку комп'ютерних класів та придбання ліцензійного програмного забезпечення;
- мінімальна кількість необхідних ліцензій за рахунок їх централізованого використання;
- якісно інший рівень отримання сучасних знань – студенти отримують можливість знаходитися в процесі навчання в будь-який час та у будь-якому місці, де є підключення до Інтернету;
- більш ефективний інтерактивний навчальний процес;
- можливість швидкого створення, адаптування й тиражування освітніх сервісів;
- можливість для студентів здійснювати зворотній зв'язок з викладачем шляхом оцінки й коментування;
- гарантія ліцензійної чистоти програмного забезпечення, що використовується в процесі навчання;
- централізоване адміністрування програмних та інформаційних ресурсів, що використовуються в освітньому процесі.

Хмарні технології пропонують навчальним закладам нові можливості для надання динамічних і актуальних, заснованих на інтернет-технологіях додатків для електронного навчання. Дані технології несуть з собою нові ризики, але й можливості для навчальних закладів і тих, які навчаються, відповідно, надавати й отримувати кращі сервіси за менші гроші. Вони можуть використовуватися в різних формах і на різних рівнях навчального процесу. За допомогою хмарних ресурсів можна сформувати інформаційно-навчальне середовище, яке може продуктивно використовуватись при виконанні творчих завдань, служити основою для організації самостійної роботи студентів.

Для забезпечення адресної роботи зі студентами та налагодження якіснішої роботи усіх інформаційних сервісів під час організації освітнього процесу доречно використовувати єдину систему створення та підтримки акаунтів користувачів інформаційної екосистеми навчального закладу з підтримкою поштових адрес у домені закладу для усіх учасників освітнього процесу (студентів, викладачів, співробітників). Разом з єдиною системою аутентифікації та авторизації це також надає можливість користуватися зовнішніми хмарними ресурсами, використовуючи OpenId та OAuth. В ХНУРЕ у травні 2015 р. прийнято рішення використовувати сервіси Google Apps for Education (GAFE).

Сьогодні GAFE пропонує повний спектр всіх наявних сервісів безкоштовно для навчальних закладів. У перелік найпопулярніших можна зарахувати Google Drive, Docs, Sheets, Slides, Mail, Search, YouTube. Більшості користувачів ці сервіси вже відомі з персонального вжитку, але GAFE надає більше можливостей корпоративним користувачам. І навіть більше, повна інтеграція сервісів створює унікальну екосистему і єдине сховище документів і файлів, що не має обмежень, тоді як для побутових користувачів є обмеження на обсяг зберігання інформації. Перелічені сервіси з успіхом замінюють більшість традиційно вживаних офісних додатків. Користувачі, що

спробували сервіси Google, згодом видаляють зі своїх коп'ютерів ці додатки і надалі користуються сервісами Google.

Крім основних сервісів, Google пропонує низку не менш важливих сервісів, що підвищують якість спільної роботи та надають можливості одночасного створення інформаційних компонентів навчального процесу. До таких сервісів необхідно додати Google Groups, G+, Hangouts, Calendar, Contacts, Forms й Sites. Кожен з цих сервісів має свою функціональність, хоча дуже часто функції перекривають одна одну.

Крім цього, забезпечена тісна інтеграція сервісів Google з системою управління дистанційним навчанням – LMS Moodle. Підтримка способів проведення самостійної роботи студентів може здійснюватися інструментальними засобами LMS Moodle. Для створення тих чи інших елементів підтримки самостійної роботи передбачається обов'язкова наявність відповідного модуля дисципліни (дистанційного курсу) на сайті університетського сервісу дистанційної освіти, що функціонує на базі Moodle.

Викладач може використовувати цей курс не тільки як портал для концентрації інформації за його дисципліною, але також і для організації усіх видів самостійної роботи.

Інтегрування системи MOODLE з хмарними сервісами забезпечується через репозиторії, доступ до яких здійснюється через панель адміністратора і розділи меню «Керування сайтом/Плагіни/Репозиторії». Репозиторії надають можливість користувачам завантажувати файли різних типів і форматів із різних зовнішніх сховищ. Перелік активованих репозиторіїв системи MOODLE доступний через панель адміністратора і розділи меню «Керування сайтом/Плагіни/Огляд плагінів».

Для забезпечення online-консультацій, проведення вебінарів та online-лекцій в системі MOODLE можна використовувати безкоштовний сервіс від компанії Google Aps – Hangouts On Air, інтеграція якого з MOODLE описана на офіційному сайті Moodle. Hangouts On Air надає можливість викладачам здійснювати відео-трансляцію в режимі реального часу, при цьому створений канал може спостерігати необмежена кількість пасивних слухачів і брати безпосередню активну участь у трансляції до 10 слухачів. Для роботи з Hangouts On Air необхідно налаштувати канал YouTube і аккаунт в Google+, які повинні бути пов'язані між собою. Під час трансляції відбувається запис, який потім можна переглянути на YouTube і в Google+.

Крім хмаро-орієнтованих сервісів, за допомогою яких можна створювати та зберігати навчальні матеріали, можна використовувати хмаро-орієнтовані сервіси, в яких зберігаються вже розроблені іншими користувачами навчальні матеріали, програми, симулятори тощо. Як правило для підключення таких сервісів потрібно додати відповідний модуль до системи управління навчальними матеріалами MOODLE.

У сучасній українській освіті, як і у всьому світі, велика увага приділяється самостійній роботі студентів з вивчення додаткових матеріалів, що стосуються змісту дисциплін, які вивчаються в аудиторії. Велика кількість різноманітних джерел інформації передбачає постійний доступ і викладачів, і студентів до Інтернету. Тому створення віртуальних освітніх просторів навчальних закладів, у яких студенти можуть отримувати якісні навчальні матеріали та консультації, є вельми важливим кроком у перетвореннях, що відбуваються у сучасній освіті.

Висновок. Здійснення самостійної роботи з використанням інформаційного освітнього середовища сприяє формуванню самостійності студентів, сприяє засвоєнню знань, формуванню вмінь, навичок за умови запровадження інтеграції традиційних та інформаційно-комунікативних технологій навчання.

Використання сучасних інформаційно-комунікаційних технологій дозволяє не тільки самостійно працювати студентам, а й організувати ефективний контроль за їх самостійною роботою. Систематичний контроль є необхідною умовою підвищення ефективності самостійної роботи студентів і, як наслідок, вдосконалення якості підготовки фахівців.

Зазначені інструментальні засоби підтримки самостійної роботи студентів на основі LMS Moodle з використанням хмарного сервісу Google Apps for Education дають змогу студентам, які у територіальному або у часовому вимірі не можуть відвідувати усі заняття в навчальному закладі, дотримуватись графіка навчального процесу, дають змогу оптимальніше організувати свій навчальний час.

Ефективність використання інформаційно-комунікаційних технологій в організації самостійної роботи студентів у багатьох випадках залежить від успішного розв'язання задач методичного характеру, що пов'язані з їх використанням в освітньому процесі.

1. Биков В. Ю. Хмарні технології, ІКТ-аутсорсинг і нові функції ІКТ підрозділів освітніх і наукових установ / В. Ю. Биков // Інформаційні технології в освіті. – №10. – 2011. – с. 8-23.
2. Бойко Н.І. Організація самостійної роботи студентів вищих навчальних закладів в умовах застосування інформаційно-комунікаційних технологій: Дис. канд. пед. наук: 13.00.04 / Нац. пед. ун-т ім.М.П.Драгоманова. – К., 2007. – 239 с.
3. Гуревич Р.С., Кадемія М.Ю. Інформаційно-телекомунікаційні технології в навчальному процесі та наукових дослідженнях: навчальний посібник для студентів педагогічних ВНЗ і слухачів інститутів післядипломної педагогічної освіти. – Київ: Освіта України, 2006. – с. 249.
4. Григулич С. М. Планування самостійної роботи / С. М. Григулич, В. О. Швець // 36 наук. праць Національний педуніверситетім. М. П. Драгоманова. – К., 2005. – С. 23–26.
5. Костікова І. І. Формування культури інформаційно-комунікаційних технологій студентами [Електронний ресурс] / І. І. Костікова. – Режим доступу: <http://eprints.zu.edu.ua/4569/>
6. Франчук В.М., Микитенко П.В. Використання Open Source Physics у LCMS Moodle : [Електронний ресурс] / В.М. Франчук, П.В. Микитенко // Інформаційні технології і засоби навчання. – 2015. – № 1 (45). – С. 156-168. – Режим доступу до журналу: <http://journal.iitta.gov.ua/index.php/itlt/article/view/1187/899#>.

УДК 004.04:004.92

Коцюба А.Ю., Сітовський В.О.

Луцький національний технічний університет

РЕАЛІЗАЦІЯ ТЕОРІЇ КЛІТИННИХ АВТОМАТІВ В ПОПІКСЕЛЬНІЙ ОБРОБЦІ РАСТРОВОЇ ГРАФІКИ

Коцюба А.Ю., Сітовський В.О. Реалізація теорії клітинних автоматів в попиксельній обробці растрової графіки. В статті розглянуто теорію абстрактних автоматів (зокрема, теорію клітинних автоматів) та описано основні аспекти її реалізації в програмуванні на прикладі технології попиксельної обробки графіки, що базується на понятті клітинного автомата. Обґрунтовано доцільність використання та впровадження розроблених продуктів на основі теорії клітинних автоматів в середовищі CodeGearRADStudioC++ Builder 2009 для вирішення поставлених цілей та завдань.

Ключові слова: C++ Builder, попиксельна обробка графіки, клітинний автомат.

Коцюба А.Ю., Ситовский В.А. Реализация теории клеточных автоматов в попиксельной обработке растровой графики. В статье рассмотрена теория абстрактных автоматов (в частности, теория клеточных автоматов) и описаны основные аспекты её реализации в программировании на примере технологии попиксельной обработки графики, основанной на понятии клеточного автомата. Обоснована целесообразность использования и внедрения разработанных продуктов на основе теории клеточных автоматов в среде CodeGearRADStudioC ++ Builder 2009 для решения поставленных целей и задач.

Ключевые слова: C++ Builder, попиксельная обработка графики, клеточный автомат.

Kotsiuba A.Yu, Sitovskiy V.O. Realization of the theory of cellular automata for pixel graphics processing. The theory of abstract automata (in particular, the theory of cellular automata) is considered in the article. It describes the main aspects of its implementation in programming using the example of pixel graphics processing technology, based on the concept of a cellular automaton. The expediency of using and implementing the developed products based on the theory of cellular automata in the CodeGearRADStudioC ++ Builder 2009 environment is substantiated for solving goals and objectives set.

Keywords: C++ Builder, pixel graphics processing, cellular automaton.

Вступ. На часі однією з найбільш перспективних є технологія подійно-орієнтованого програмування, яка базується на використанні математичного апарату теорії абстрактних автоматів. Особливо значних результатів можна досягнути, поєднуючи цю технологію з іншими відомими – наприклад, з технологією об'єктно-орієнтованого програмування. Це поєднання технологій має широкий спектр застосування в різних галузях знань: від астрономічних фотографій, медицини та робототехніки, і закінчуючи контролем якості в промисловості. Слід зазначити, що теорія абстрактних автоматів передбачає великий різновид класів автоматів і найбільш використовуваними в даному напрямку серед них є клітинні автомати. Однією із найвідоміших класичних реалізацій даного математичного апарату в програмуванні є гра "Життя". Вона реалізована за принципом ітераційної зміни кольору в клітинках (з чорного на білий або навпаки) в залежності від кольорів в сусідніх клітинках. Виникає ідея розвитку даного алгоритму в напрямках: замість клітинки будемо брати окремий піксель растрового зображення; замість двох кольорів використаємо палітру RGB, у якій налічується 256^3 кольорів. Якщо реалізувати цю ідею в напрямку обробки цифрових зображень, то з'явиться можливість вирішувати широкий клас практично значимих задач пов'язаних з теорією обробки цифрових зображень.

Проблема яку слід розв'язати полягає у тому що, розроблений в середовищі C++ Builder програмний продукт обробки растрових зображень, який базується на понятті клітинного автомата повинен вміти зчитувати вхідну інформацію двома способами: за допомогою функціональної залежності, яка формує растрове зображення та за допомогою відкриття уже готового зображення в різних форматах. Далі в залежності від кольору в сусідніх пікселях повинні змінюватись кольори у всіх пікселях вхідного зображення. Цей процес можна ітеративно повторювати будь-яку кількість разів. При цьому слід реалізувати автоматичне збереження всіх результуючих зображень у вибраному користувачем форматі.

Для вирішення поставлених завдань та реалізації вищеописаного підходу в алгоритмах обробки растрових зображень розробимо в середовищі візуального програмування CodeGearRAD-Studio C++ Builder 2009 програмний продукт і забезпечимо у ньому виконання наступних функцій:

- завантаження растрового зображення;
- перетворення зображення у двоколірне;
- обробки зображення за допомогою алгоритму усереднення з вагою;
- обробки зображення за допомогою алгоритму, який порівнює кожен піксель з сусідами, і в разі подібності замінює його на подібний;

- обробки зображення за допомогою алгоритму виключення хибних пікселів;
- обробки двоколірного зображення за алгоритмом, запропонованим у класичній грі "Життя";
- автоматичне чи ручне збереження результату обробки на будь-якому етапі роботи.

Для виконання вищевказаних функцій служать основні алгоритми:

- алгоритм, що допомагає користувачу вибирати різні два кольори так, щоб один з них був темним, а другий світлим;
- алгоритм попиксельного зчитування зображення;
- алгоритм збереження користувачем результуючого зображення у будь-якому з допустимих діалогом збереження форматі;
- алгоритм формування зображення за допомогою функції від координат пікселя $f(i,j) \rightarrow \rightarrow TColor$;
- вищеописані алгоритми обробки зображення.

Аналіз досліджень в напрямку обробки растрового зображення показує, що на сьогоднішній день таких алгоритмів та середовищ їхньої реалізації існує величезна кількість. Розроблений в даній роботі проект не призначений для конкуренції з ними, а його **метою** є лише отримання навиків та знань в напрямку створення повноцінних візуальних програмних додатків, за допомогою яких в перспективі можна було б, наприклад, наочно та зрозуміло для студентів проводити апробацію та модернізацію відомих в літературі методів обробки графіки.

Термін "**клітинні автомати**" (КА) почав використовуватись у середині ХХ ст. для позначення сукупності залежних елементів із заданими станами і правилами, за допомогою яких стани цих елементів і залежності між ними змінюються в часі. Час і стани при цьому дискретні. Використання описаних моделей для формального моделювання самовідтворюваних організмів вперше запропоновано в роботі Фон Неймана. Елементи клітинних автоматів запропоновано представити одновимірними або двовимірними нескінченними прямокутними таблицями. Стан елемента змінюється в залежності від його стану і від стану найближчих сусідів.

Класичні клітинні автомати в загальному випадку відповідають наступним критеріям:

- зміна значень всіх клітинок відбуваються одночасно після обчислення нового стану кожної клітинки решітки. Інакше порядок перебору клітин решітки при проходженні ітеративного процесу суттєво впливав би на результат;

- решітка однорідна;
- взаємодії локальні. Лише околишні клітинки (як правило, сусідні) здатні вплинути на дану клітинку;

- множина станів клітинки скінченна. Ця умова потрібна, щоб для отримання нового значення стану клітини треба було виконати скінченну кількість операцій (але це не заважає використовувати клітини для зберігання чисел із плаваючою комою для розв'язку прикладних задач).

Розроблений програмний продукт складається з: 3-х модулів (форм: головної, функціонального та ітераційного налаштування). Інтерфейс користувача досить простий та інтуїтивно зрозумілий. Для скорочення викладок про процес розробки та етапи користування опишемо основні можливості програми за допомогою чотирьох прикладів. Першим прикладом побудуємо двоколірне зображення на основі "реальної" (тут мається на увазі – фото власного виробництва) фотографії розмірністю 1760x1060, зображеної на рис. 1.



Рисунок 1 – Вигляд вхідної фотографії розмірністю 1760x1060

Приклад 1. Спочатку відкріємо дане фото за допомогою розробленої нами програми. Далі натискаємо кнопку, на якій зображено чорно-білі нігті, і вибираємо два кольори, як це описано на рис. 2.

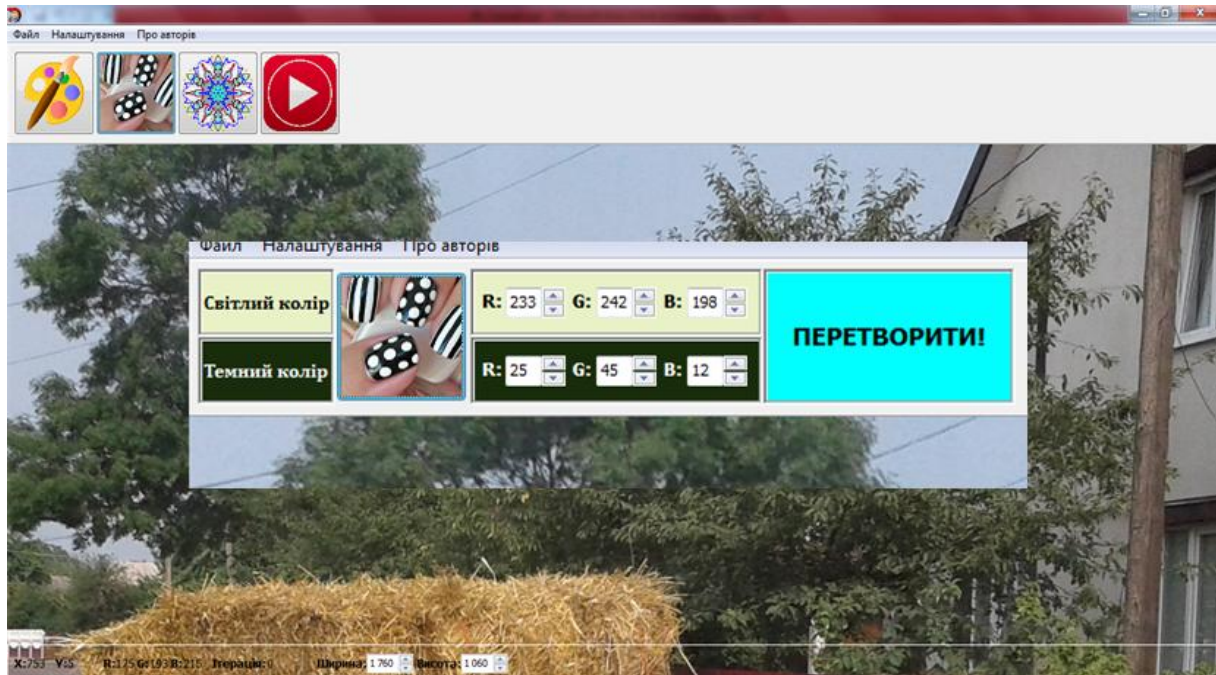


Рисунок 2 – Налаштування двох кольорів для перетворення зображення у двоколірне

Зауважимо, що при виборі кольорів програма відслідковує, щоб темний колір не став світлим, а світлим – відповідно темним. Результат обробки збережемо та подамо на рис. 3.



Рисунок 3 – Двоколірний результат обробки фотографії розмірністю 1760x1060

Очевидно, що таких результатів неважко досягнути за допомогою відомих растрових редакторів, але в даному програмному проєкті нам вигідно було реалізувати власну вищеописану обробку, щоб далі на отриманому результаті перевіряти різні алгоритми, які розраховані в першу чергу на два кольори (найвідоміший серед таких алгоритмів є класичний алгоритм гри "Життя"). Оскільки обробка такої розмірності займає близько 10 хв. (а даний алгоритм перетворення досить примітивний), то надалі обмежимося значно меншою розмірністю.

В наступному прикладі запропоновано розглянути дві функціональні заливки.

Приклад 2. Покажемо як зробити заливку фону за допомогою функції $f(i,j) \rightarrow TColor$ (Налаштування-Функціональне налаштування). Розглянемо дві заливки: випадкову і таку, що задається функцією, яку подано на рис. 4.

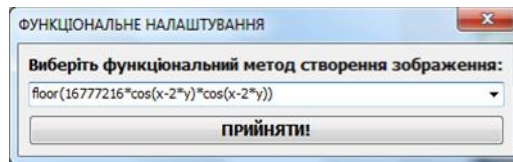


Рисунок 4 – Вигляд діалогового вікна “Функціональне налаштування”

За умовчанням застосовується генератор псевдовипадкових чисел (ГПВЧ), тобто для кожного пікселя з множини потужністю 2^{24} всіх можливих кольорів програма сама вибирає потрібний. Очевидно, що результат є нестабільним і в нашому випадку матиме вигляд як на рис. 5 ліворуч. А на рис. 5 праворуч покажемо результат заливки, що задається функціональним налаштуванням на рис. 4.

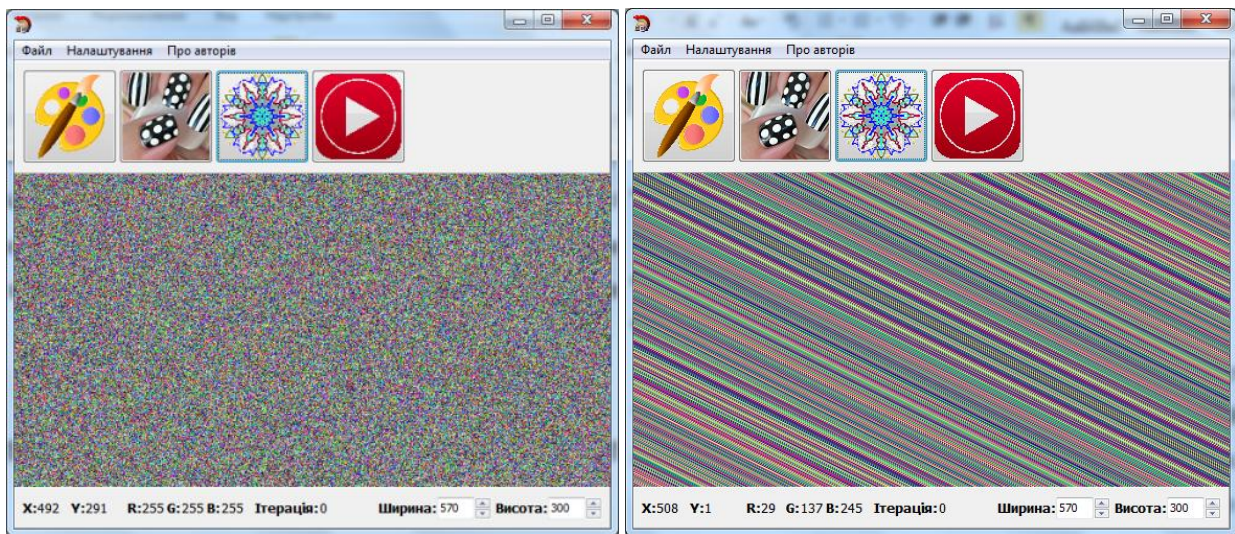


Рисунок 5 – Приклади заливок фону

Очевидно, що пошук різного роду заливок, зводиться до експериментів, які обмежуються лише математичною фантазією.

В наступних прикладах покажемо результати деяких алгоритмів обробки зображень.

Приклад 3. Спочатку розкажемо, як користуватися алгоритмом усереднення з вагою. Для цього приймемо, що в ітераційному налаштуванні були встановленні значення параметрів, заданих в ітераційному налаштуванні (рис. 6).

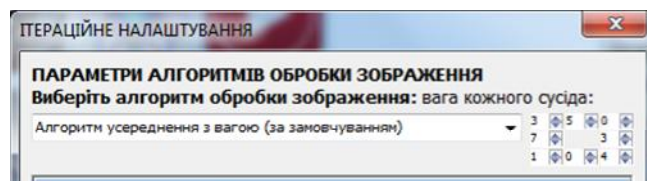
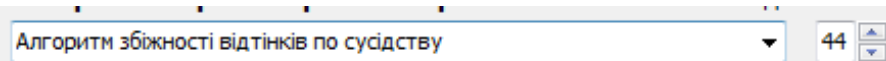


Рисунок 6 – Вигляд діалогового вікна “Ітераційне налаштування”

По-суті все зводиться до підрахунку середнього арифметичного з вибірки, у якій кожен елемент може зустрічатися декілька разів. Зауважимо, що випадок, коли кожна вага є нульовою, не зашкодить роботі алгоритму, просто у цьому випадку програма навіть не почне працювати і видасть повідомлення про помилку.

На рис. 7 показано результат після декількох ітерацій обробки зображення за допомогою алгоритму усереднення з вагою при заданих нами значеннях параметрів для випадково вибраної фотографії.

Далі опишемо роботу



і покажемо, як він працює при вказаному нами максимальному відхиленні “44”. І аналогічно, як і у попередньому випадку, проведемо декілька ітерацій та отримаємо результат (див. рис. 8).

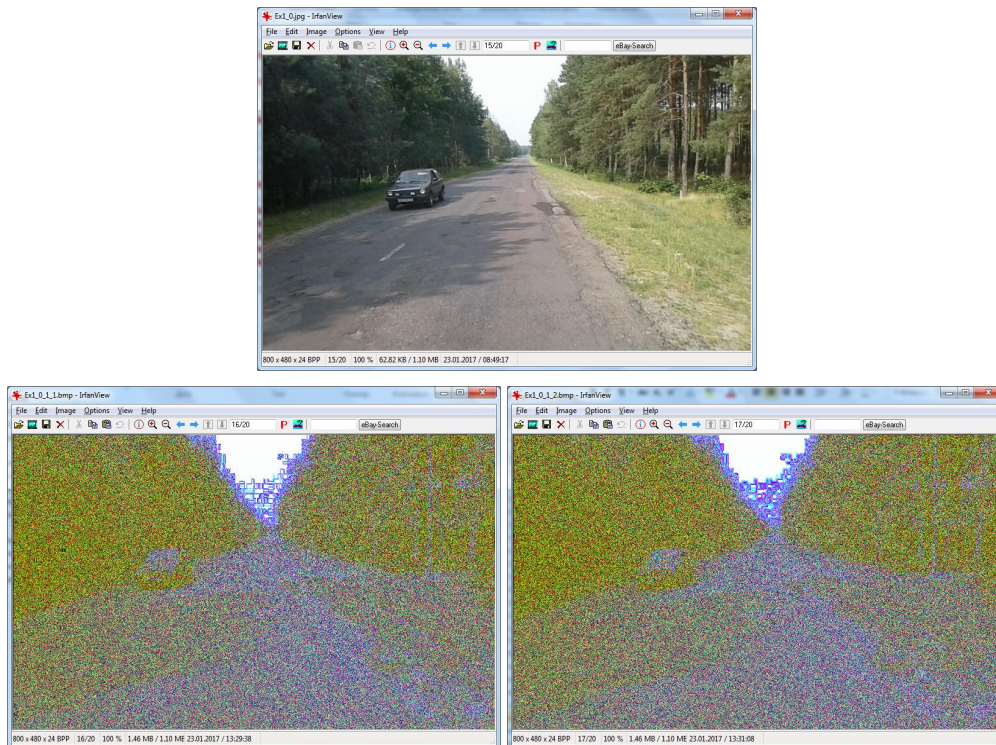


Рисунок 7 – Результати обробки перших двох ітерацій алгоритму усереднення з вагою

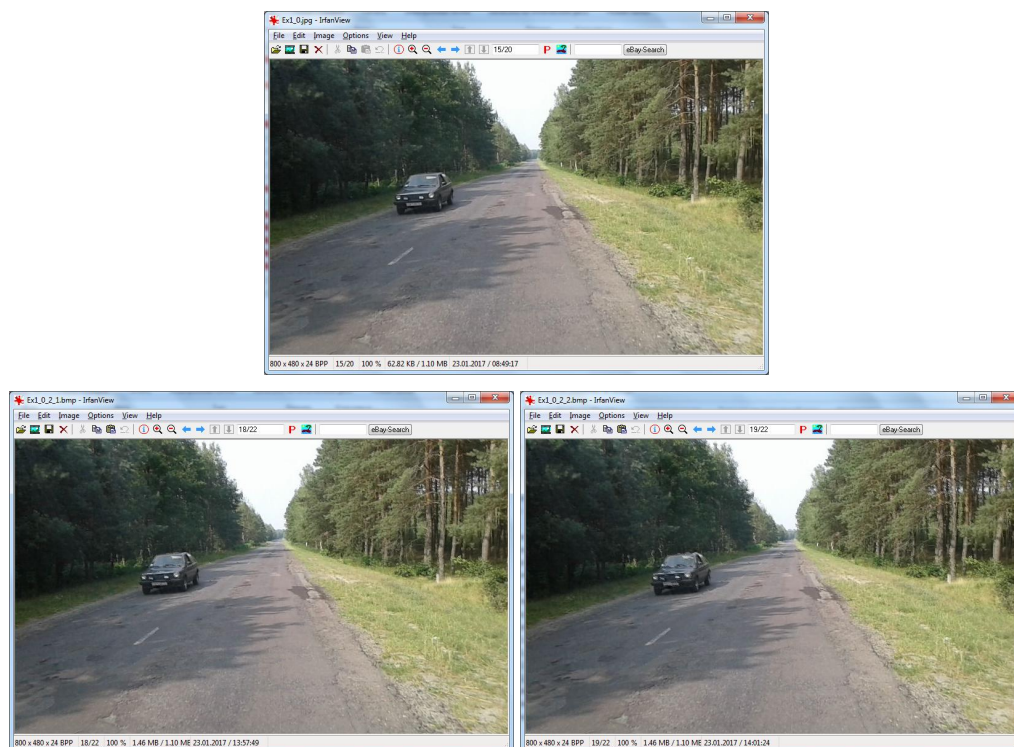


Рисунок 8 – Результати обробки перших двох ітерацій алгоритму збіжності відтінків

Очевидно, що другий алгоритм обробки повільніше спотворює зображення, ніж перший. Виникає питання: для чого потрібно спотворювати фотографію. На перший погляд питання доречне. Але як створювати обернені алгоритми, якщо не буде можливість перевірки. Складання оберненого алгоритму це складна задача, яка зводиться до розв'язування системи великої кількості здебільшого нелінійних (але не для випадку усереднення) рівнянь або нерівностей з великою кількістю невідомих. З точки зору сучасних технологій – це складна проблема, але не на стільки, щоб її не можна було би вирішити.

Приклад 4. В останньому прикладі зупинимося на алгоритмі, який буде знищувати спотворення зображення у вигляді хибних пікселів. Назвемо його алгоритмом знищення хибних пікселів, що мають групи з подібних сусідів. По суті цей алгоритм шукає такі пікселі, з обох сторін, якого є скупчення близьких за кольором сусідів. При цьому колір цього пікселя повинен відрізнятися від кольорів груп сусідів (як правило це дві групи по 3 або 4 пікселя, розташованих поруч), а кожна з цих груп може бути “ворожою” не лише до нашого пікселя, але і між собою. Щоб показати принцип роботи даного алгоритму достатнім буде детальний аналіз лише однієї ітерації. Аналогічно, як і в другому алгоритмі, тут необхідно встановити максимальне відхилення. Нехай воно є рівним “15”. Покажемо спочатку на рис. 9 вхідне зображення, інвертовану різницю за абсолютною величиною між вихідним та вхідним файлами, а потім уже результат обробки, який візуально майже не відрізняється від вхідного файлу.

Найбільш цікавим є той факт, що даний алгоритм майже нічого не спотворює, а навпаки – досить акуратно знаходить проблемні пікселі і виправляє. І код цього алгоритму найскладніший, а код першого – відповідно найпростіший. Тобто виходить що, для того, щоб швидко змарнувати зображення великої праці не треба, а виправляти помилки набагато важче.

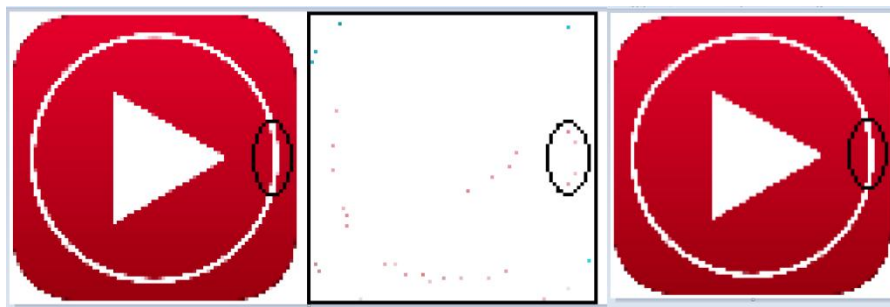


Рисунок 9 – Результати обробки зображення алгоритмом знищення хибних пікселів, що мають групи подібних сусідів

Висновки. Отже, розроблений нами програмний продукт наочно показує як можна реалізувати теорію клітинних автоматів для попиксельної обробки графіки, а саме автоматизованої обробки растрового зображення за допомогою програмування. Хоча кількість методів обробки є обмеженою, але можливим є додавання будь-якого сучасного відомого алгоритму перетворення графіки (наприклад, за допомогою нашої програми у навчальному процесі можна показувати результати всіх обробок зображення, що пропонуються студентам в курсі таких дисциплін: “Стеганографія”, “Обробка цифрових сигналів та зображень” тощо). І це дає можливість далі продовжувати роботу в даному напрямку. Щодо використання розробленого продукту у не навчальних цілях, то це може бути, наприклад, таке перетворення зображення, яке підвищує його якість: знімає розмитість, видаляє зайве (поодинокі шуми – пікселі).

Поставлені проблеми були повністю виконані і є значна перспектива в розширенні зробленого програмного продукту, що вимагає нових досліджень.

1. Архангельский А.Я. Программирование в С++ Builder / А.Я. Архангельский. – М. : ООО “Бином-Пресс”, 2010. – 896 с.
2. Архангельский А.Я. Язык С++ в С++ Builder. Справочное и методическое пособие / А.Я. Архангельский. – М. : ООО “Бином Пресс”, 2007. – 1012 с.
3. Вельдер С.Э. О верификации простых автоматных программ на основе метода Model Checking / С.Э. Вельдер, А.А. Шальто // Программные и аппаратные средства. – 2007. – № 3. – С. 27-38.
4. Капітонова Ю.В. Основы дискретной математики / Ю.В. Капітонова, С.Л. Кривий, О.А. Летичевський та ін. – К. : “Наукова думка”, 2002. – 580 с.
5. Карпов Ю.Г. Теория автоматов / Ю.Г. Карпов. – СПб. : “Питер”, 2002. – 206 с.
6. Культин Н.Б. С++ Builder в задачах и примерах / Н.Б. Культин. – СПб. : “БХВ-Петербург”, 2005. – 336 с.
7. Пахомов Б.И. Самоучитель C/C++ и С++ Builder 2007 / Б.И. Пахомов. – СПб. : “БХВ Петербург”, 2008. – 672 с.
8. Поликарпова Н.И. Автоматное программирование / Н.И. Поликарпова, А.А. Шальто. – СПб. : “Питер”, 2009. – 167 с.
9. Понимание ООП в JavaScript [Електронний ресурс]. – Режим доступу : <http://habrahabr.ru/post/-153365>.
10. С++ Community [Електронний ресурс]. – Режим доступу : <http://www.cplusplus.com>.
11. UniMod: Инструмент для построения схем автоматов в формате UML и генерации по ним исходного кода [Електронний ресурс]. – Режим доступу : <http://unimod.sourceforge.net>.

УДК 004.312.26:519.725

Крайник Я.М., Перов В.О.

Чорноморський національний університет імені Петра Могили

ОРГАНІЗАЦІЯ СКІНЧЕНОГО АВТОМАТУ УПРАВЛІННЯ ПРОЦЕСОМ ДЕКОДУВАННЯ ДЛЯ ДЕКОДЕРУ TURBO-PRODUCT-КОДІВ НА БАЗІ FPGA

Крайник Я.М., Перов В.О. Організація скінченного автомату управління процесом декодування для декодера Turbo-Product-кодів на базі FPGA. У даній статті запропонований метод організації скінченного автомату для управління процесом декодування Turbo-Product-кодів на базі FPGA. Результуючий скінчений автомат використовує діапазони значень лічильника для реалізації під-станів скінченного автомату. Запропонований метод організації автомату для управління декодером дозволяє підвищити пропускну здатність декодера, оскільки виконує обробку вхідних даних відповідно до довжини кодів за необхідну кількість тактів.

Ключові слова: скінчений автомат, Turbo-Product-коди, реконфігуровний декодер, діапазони, періоди активності.

Крайник Я.М., Перов В.О. Организация конечного автомата управления процессом декодирования для декодера Turbo-Product-кодов на базе FPGA. В данной статье предложен метод организации конечного автомата для управления процессом декодирования Turbo-Product-кодов на базе FPGA. Результирующий конечный автомат использует диапазоны значений счетчика для реализации под-состояний конечного автомата. Предложенный метод организации автомата для управления декодером позволяет повысить пропускную способность декодера, поскольку выполняет обработку входных данных в соответствии с длиной кода за необходимое количество тактов.

Ключевые слова: конечный автомат, Turbo-Product-коды, реконфигурируемый декодер, диапазоны, периоды активности.

Krainyk Y.M., Perov V.O. Organization of Finite-State Machine for Decoding Process Control in Turbo-Product-Code Decoder based on FPGA. In this article method for organization of finite-state machine for decoding process control in Turbo-Product-code decoder based on FPGA has been proposed. The finite-state machine utilizes ranges of counter values to implement sub-states. The devised finite-state machine organization method allows improving in decoder throughput, as it performs processing of input data according to the code length and uses necessary number of clock pulses.

Keywords: Finite-State Machine, Turbo-Product-codes, reconfigurable decoder, ranges, activation periods.

Завадостійкі коди є ключовим компонентом систем високошвидкісної передачі інформації. Саме від властивостей таких кодів залежить, наскільки надійною буде передача даних та яка пропускна здатність можлива для реалізації певного інтерфейсу. Серед корегуючих кодів наявні коди, які забезпечують різні характеристики щодо корегуючої здатності. Їх вибір для використання у системі залежить від заданих критеріїв, що оцінюють середовище передачі сигналу та вплив зовнішніх факторів на даний процес. Коди з низькою щільністю перевірки на парність, коди Хеммінга, турбо-коди (англ. Turbo-Product Codes - TPC) демонструють різну пропускну здатність і відрізняються за складністю реалізації та вимогами щодо апаратних ресурсів. Тому кінцевий вибір для їх використання у системах передачі інформації у більшості випадків є компромісом між характеристиками коду та доступними ресурсами.

Програмовні логічні інтегральні схеми (ПЛІС, англ. Field-Programmable Gate Array - FPGA) є основним засобом для відлагодження роботи інтерфейсу передачі даних. Подальшим кроком для створення готових пристроїв є перенесення протестованого рішення на спеціалізовані мікросхеми. З іншого боку, використання ПЛІС у кінцевому пристрої може мати свої переваги, які полягають у можливості зміни налаштувань, удосконаленні початкової розробки, і, навіть, повній її переробці. З точки зору використання ПЛІС у системах передачі інформації, то реалізований з їх допомогою функціонал повинен забезпечувати певний резерв щодо пропускну здатності, відповідно, мати ресурси щодо розширення засобів для передачі даних. Тому при проектуванні рішення для системи високошвидкісної передачі інформації, яка базується на мікросхемах ПЛІС, необхідно передбачити використання мікросхеми, яка здатна забезпечити нові вимоги протягом кількох циклів розробки системи.

Декодер завадостійких кодів є ключовим компонентом у системі передачі інформації, оскільки саме на нього припадає найбільша кількість операцій по обробці даних. Реалізація апаратного декодування Turbo-Product-кодів (TP-кодів) [1] передбачає наявність кількох стадій декодування: запис вхідних даних, декодування рядків, декодування стовпців, видача результату та ін. проміжні стадії. Керування виконанням окремих стадій доцільно організувати з використанням технік побудов скінченного автомату. Керуючий автомат для декодера, який може працювати з кодами довільної довжини без необхідності додаткового перепрограмування

(реконфігуровний декодер) повинен забезпечувати обробку вхідних даних на всіх стадіях декодування і при цьому виконувати кожен окрему стадію за мінімально необхідну кількість тактів. Існуючі рішення [2], в основному, орієнтовані на обробку одного коду або кодів, які мінімально відрізняються між собою. Тому проблема побудови скінченного автомата для управління процесом декодування ТР-кодів є актуальною та важливою для задачі реалізації реконфігуровного декодера ТР-кодів на базі програмовних логічних інтегральних схем (ПЛІС) [3, 4].

Проблемі декодування ТР-кодів присвячена велика кількість наукових досліджень, серед яких [5-8]. Зазвичай, у цих роботах акцент робиться на алгоритмах декодування. При цьому, складність обчислень, які необхідно виконувати, не оцінюється відносно переносу на конкретну платформу для реалізації. Тим не менш, це є ключовим питанням при реалізації алгоритмів для FPGA, оскільки дані мікросхеми мають особливості, які не можуть бути не враховані (обмежений обсяг внутрішньої пам'яті, відсутність або невелика кількість обчислювачів, що дозволяють проводити операції з плаваючою точкою та ін.).

Ключовим параметром, який характеризує роботу декодера завадостійких кодів, є пропускна здатність. Пропускна здатність реконфігуровного декодера в залежності від того, як організоване управління процесом декодування може значно відрізнитись для різних кодів. У відомих дослідженнях обробці кодів з різною структурою присвячено недостатньо уваги, тому це питання потребує подальших досліджень і є актуальним.

Виклад основного матеріалу. Для реалізації керуючих блоків у складі FPGA для систем різного типу найбільш поширеним варіантом є реалізація на основі скінченного автомату (англ. Finite State Machine - FSM). Саме з використанням даного підходу має бути розроблений блок керування блоками декодування всередині декодера. Проте, у наукових джерелах [5-8], наводиться опис архітектури декодера без уточнень з приводу того, яким чином здійснюється керування. Даний момент є важливим, оскільки декодер повинен забезпечувати достатній рівень паралельності процесу декодування для отримання необхідної пропускної здатності. Тому невирішеною є проблема організації скінченного автомату для керування процесом декодування ТР-кодів.

Скінчений автомат передбачає керування декодером [3, 4], який має архітектуру, яка в загальному вигляді представлена на рис. 1. Декодер містить блоки пам'яті (вхідний, вихідний та проміжний), пристрої зсуву для забезпечення коректної обробки рядків та колонок, безпосередньо самі блоки декодування.

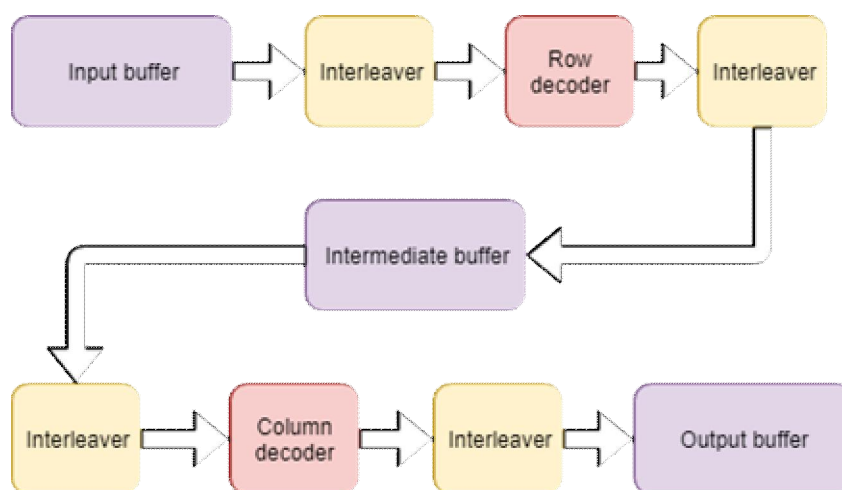


Рисунок 1 – Загальна архітектура декодера ТР-кодів

Для демонстраційних цілей деякі зв'язки між блоками у декодері були опущені. Наприклад, на другій та наступних ітераціях зчитування відбувається не з вхідного буферу, а з додаткової проміжної пам'яті.

Основними вимогами до розроблюваного автомату є простота та гнучкість налаштувань. Тобто, він повинен містити якомога менше станів, але, при цьому, забезпечувати можливість контролю для усіх стадій декодування. Розроблений автомат представлено на рис. 2.

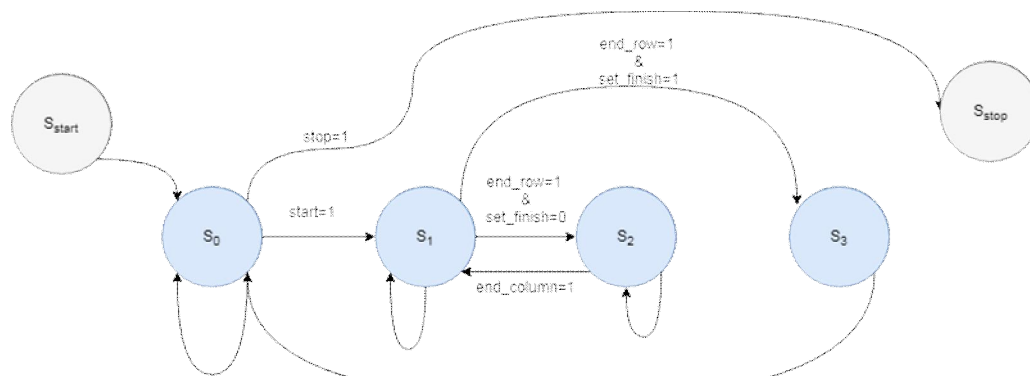


Рисунок 2 – Граф станів скінченного автомату для декодери TR-кодів

Особливістю автомату є те, що вихід з процесу декодування і перехід до видачі результату відбувається зі стану S_1 . Це пов'язано з тим, що запис у вихідну пам'ять необхідно виконувати лише у випадку, якщо значення змінної `set_finish` дорівнює "1". На момент виконання другої пів-ітерації це не може бути відомим. Тому використовується підхід, коли для перезапису використовується ще одна додаткова перша пів-ітерація.

Автомат складається з 4 основних станів, а також двох службових (запуску та виходу). Під час перебування пристрою у основних станах забезпечується наступна функціональність:

- S_0 – очікування надходження сигналу запуску декодування у той момент, коли дані для декодування готові; також вихід з процесу декодування;
- S_1 – декодування рядків у матриці вхідних даних;
- S_2 – декодування стовпців у матриці вхідних даних;
- S_3 – завершення процесу декодування, видача сигналу про готовність оброблених даних до видачі.

Обробка вхідних даних за ітераціями декодування забезпечується за рахунок наявності взаємних переходів між станами S_1 та S_2 . Обробка кількох пакетів забезпечується наявністю переходу між станами S_1 та S_3 . Оскільки декодер є реконфігуровним, то передбачена також можливість виходу (S_{stop}) з робочого циклу та зупинка роботи декодера для встановлення нових параметрів роботи декодера. У даний момент часу декодер не може приймати вхідні дані або виконувати будь-яку обробку даних в основному циклі, оскільки це призведе до отримання неправильного результату.

У автоматі керування декодером для виділення під-станів використовується значення лічильника, який працює постійно під час роботи декодера. Подібний підхід дозволяє, по-перше, забезпечити максимальну гнучкість налаштувань по управлінню керуючими імпульсами, по-друге, залучити для цього мінімальну кількість додаткових ресурсів – використовується лише регістр для значення лічильника. Кожному під-стану відповідає діапазон значень лічильника. Розбиття на діапазони відбувається для таких виділених стадій декодування (послідовність застосовується і для декодування рядків і стовпчиків):

- зчитування даних з пам'яті;
- передача даних на регістр зсуву;
- безпосередньо декодування;
- передача даних для зворотного зсуву;
- запис даних.

Усі зазначені стадії пов'язані таким чином, що вихід попередньої стадії є входом для наступної. Для ввімкнення відповідного блоку на нього подається сигнал активації (сигнал `enable`). Коли сигналу активації немає, то блок не є активним і не виконує жодних операцій. Відповідно, для коректної роботи необхідно узгодити готовність даних на виходах різних блоків та активацію наступних. Декілька блоків можуть бути активними одночасно.

Розподіл значень лічильника на діапазони відбувається відповідно до наступних правил. Введемо наступні позначення для представлення процесу формування діапазонів:

- T – загальна кількість діапазонів;

- $t = \overline{1, T}$;
- $\min_t$ – ліва границя діапазону;
- $\max_t$ – права границя діапазону;
- $[\min_t, \max_t]$ – представлення діапазону t .

Ліва границя першого діапазону має значення 0, а права границя останнього діапазону $t = T$ має значення $\max_count$. Права границя попереднього діапазону та ліва границя наступного пов'язані між собою наступним співвідношенням:

$$\min_t = \max_{t-1} + 1 . \quad (1)$$

Відповідно, діапазони не мають перетинатися між собою. Наступні параметри впливають на визначення значень діапазонів:

1. Довжина найдовшого коду n_c у групу кодів (у випадку рядків це буде довжина коду стовпців, а для стовпців – довжина коду рядків, оскільки саме таку кількість векторів необхідно обробити протягом однієї пів-ітерації).
2. Затримка для блоку обробки між отриманням даних та видачею результату (l_{block}).

Далі розглянемо формування діапазонів значень лічильника на основі періодів активностей. Діапазони визначаються таким чином, що стан усіх сигналів активації блоків є стабільним і не змінюється. Зміна стану одного з таких сигналів означає початок нового діапазону.

У загальному випадку кожному блоку для обробки та видачі результатів необхідна наступна кількість тактів

$$p_{block} = n_c + l_{block} . \quad (2)$$

Коли відомою є загальна кількість тактів, що витрачає блок на повне опрацювання і видачу результату, стає можливим узгодження періодів активностей блоків. Дані періоди є ключовими у формуванні границь діапазонів лічильника. Періоди активності блоків можуть належати різним суміжним діапазонам.

Розглянемо детальніше періоди активностей блоків та їх взаємозв'язок між собою. На початку має виконуватись стадія зчитування даних з вхідного буферу. Активація блоку наступної стадії відбувається у момент, коли з'являються перші результати обробки на виході попереднього блоку. Відповідно ліва границя другого діапазону буде обчислена як

$$\min_2 = l_{block1} . \quad (3)$$

Після цього моменту перший блок залишається активним ще протягом n_c тактів для того, щоб повністю передати результат у наступний блок. Також з цього можна зробити висновок про те, що права границя першого діапазону буде за значенням відповідати затримці, яка є на першому блоці:

$$\max_1 = l_{block1} - 1 . \quad (4)$$

З наведених вище виразів можна визначити лише три границі для перших двох діапазонів (повністю вказати перший діапазон, а також ліву границю другого) для архітектури декодера, яка передбачає послідовне з'єднання блоків між собою. Для визначення наступних границь необхідно також враховувати параметр n_c . Оскільки він дозволяє визначити, через яку кількість тактів має бути деактивований сигнал для попереднього блоку. До того, моменту, коли кількість поданих тактових імпульсів не перевищує значення p_{block1} , границі діапазонів можна визначити лише на основі затримок блоків обробки:

$$\min_{t+1} = \sum_t l_{blockt} , \quad (5)$$

$$\max_t = \sum_t l_{blockt} - 1 . \quad (6)$$

Загальна кількість тактів, яка знадобиться для виконання однієї пів-ітерації визначиться як

$$\max_count = \sum_t l_{blockt} + n_c \quad (7)$$

Після того, як кількість поданих тактів перевищила p_{block1} , як мінімум один блок має бути деактивований, що означає необхідність нового діапазону лічильника.

Розглянемо формування діапазонів для під-станів автомату на прикладі декодера, виконання пів-ітерації у якому забезпечується п'ятьма блоками, і представленим на рис. 3.

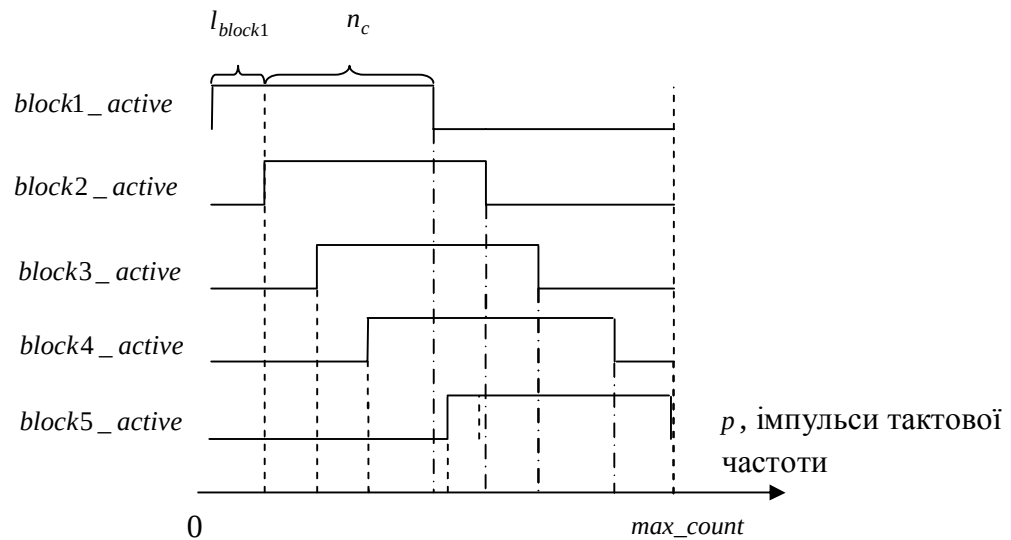


Рисунок 3 – Приклад діаграми сигналів активації блоків декодування

З даної діаграми значень сигналів підтверджується коректність виразу (7) для випадку, коли блоки з'єднані послідовно. Звертає на себе увагу той факт, що затримка останнього блоку у такому випадку не спричиняє формування нового діапазону. Крім того, на діаграмі чітко представлено розбиття на діапазони на основі періодів активностей блоків, що дозволяє визначити під-стани для запропонованого скінченного автомату. Робота реконфігуровного декодера при обробці різних кодів забезпечується таким чином, що виконується обчислення діапазонів для всіх груп кодів, для яких планується обробка даних.

На основі вхідних сигналів, які дозволяють обрати код, якому відповідає вхідна матриця даних, визначається максимальне значення, до якого необхідно вести рахунок – max_count . Зміна значення лічильника та відповідність діапазнам для представлення під-станів показана на рис. 4.

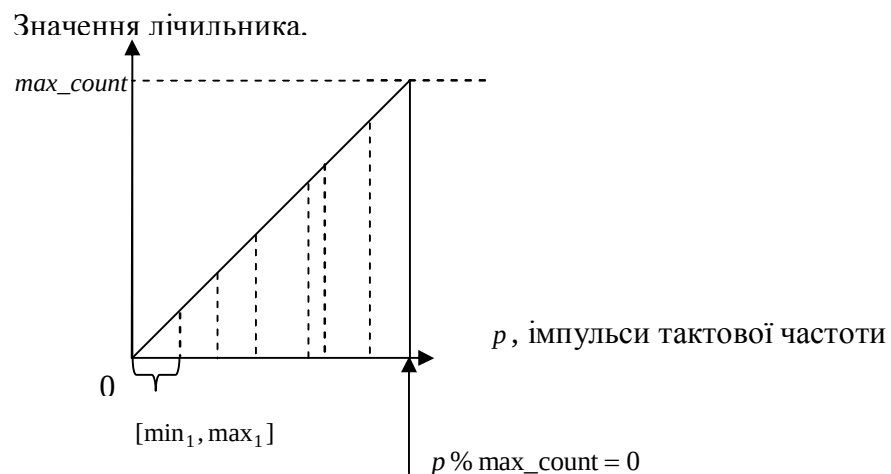


Рисунок 4 – Формування діапазонів значень лічильника

З кожним тактовим імпульсом значення лічильника збільшується на одиницю до того моменту, поки не досягне максимального значення max_count . Після цього значення обнулюється і рахунок починається спочатку.

На попередньому рисунку показано, як виглядає розподіл діапазонів для однієї пів-ітерації. На рис. 5 показано, як виглядає процес зміни лічильника протягом усього процесу декодування з акцентом на першу та останню пів-ітерації.

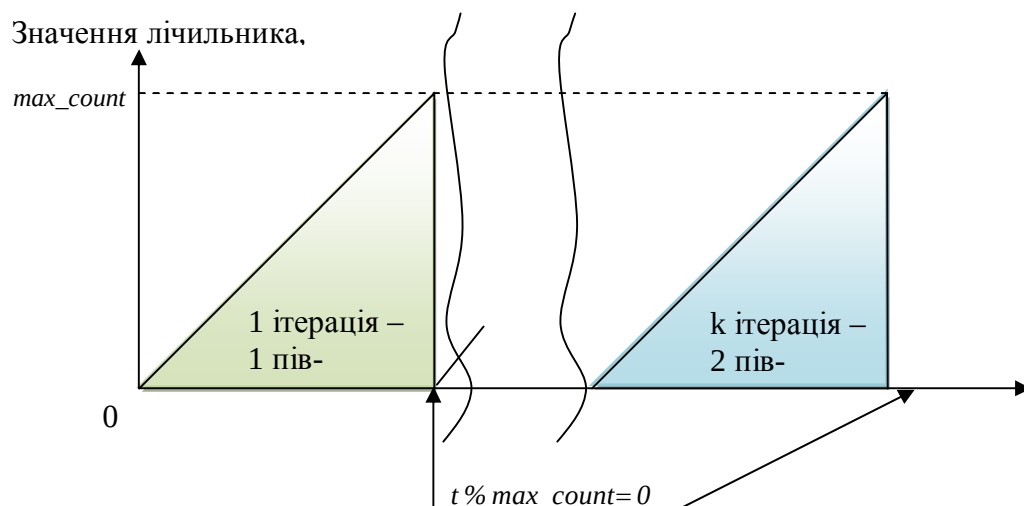


Рисунок 5 – Графік зміни значень лічильника протягом усього процесу декодування (для кодів, що відносяться до однієї групи)

На рисунку показаний випадок, коли обидва коди, що утворюють код-добуток, відносяться до однієї групи кодів. У випадку, коли для рядків та стовпців, групи кодів є різними, лічба ведеться поперемінно до різних максимальних значень. Відповідно, різні значення діапазонів будуть активними під час декодування рядків та стовпців. Цю обставину варто враховувати під час розрахунку оцінки пропускну здатності декодеру.

У випадку порівняння декодеру, який для обробки даних використовує повний цикл (і, відповідно, максимальну кількість тактових імпульсів) та декодеру, керованого запропонованим скінченим автоматом, обробка кодів з максимальною довжиною слова відбувалась би однаково, тому виграшу у пропускну здатності немає. Проте, для кодів-добутків, які базуються на кодах з довжиною меншою за максимальну стає можливим отримати виграш у пропускну здатності за рахунок того, що на обробку буде витрачатись менша кількість тактових імпульсів, аніж для декодеру з менш гнучким керуючим блоком, який використовує однаковий цикл обробки для усіх кодів.

Розроблений у даній роботі скінчений автомат, який використовує лічильник і на основі діапазонів його значень, організовує під-стани для подачі керуючих імпульсів на блоки у реконфігурованому декодері дозволяє організувати обробку даних за мінімально необхідну для обраного коду кількість тактів. Такий автомат дозволяє підвищити пропускну здатність реконфігурованого декодеру при роботі з кодами, які не відносяться до групи кодів з найбільшою довжиною кодового слова. Виграш у швидкодії залежить від максимальної довжини коду у групі, а також від довжини коду групи, для якого проводиться обробка. За рахунок того, що не обробка проводиться не для всієї матриці, а лише для необхідної частини, яка заповнена вхідними даними, вдається підвищити пропускну здатність. Відносний виграш у такому випадку можна обчислити відповідно до (7) як

$$Q = \frac{\sum_t l_{blockt} + n_{c actual}}{\sum_t l_{blockt} + n_{c top}}, \quad (8)$$

де $n_{c top}$ - довжина коду, який представляє найдовший код, який може оброблюватись декодером;

$n_{c actual}$ - довжина коду, який заданий поточною конфігурацією для обробки.

Даний показник (Q) вказує те, наскільки вдалося зменшити кількість тактів у порівнянні зі сценарієм, коли обробка проводилась з використанням ресурсів, призначених для найдовшого коду.

Висновки. Запропонований у роботі скінчений автомат дозволяє організувати роботу реконфігурованого декодеру ТР-кодів таким чином, щоб виконувати обробку повідомлень на

основі різних кодів з використанням найменшої необхідної кількості тактів. Він містить чотири основні стани, за допомогою яких здійснюється керування основними блоками обробки. Особливістю автомату у даній роботі є те, що він використовує лічильник для того, щоб реалізувати керування основними імпульсами запуску блоків обробки. Значення, які може набувати лічильник мають бути розділені на діапазони, відповідно до яких і виконуються необхідні операції. Розбиття на діапазони виконується на основі налаштувань модулів обробки даних та періодів активностей їх роботи (моменти часу, коли вони оброблюють вхідні дані). За рахунок того, що для декодування використовується мінімально необхідна кількість тактових імпульсів для конкретного коду, вдається підвищити пропускну здатність декодеру. Розроблений скінчений автомат був реалізований у складі керуючого модуля декодеру TP-кодів на базі мікросхеми ПЛІС виробництва фірми Altera, яка була надана для проведення досліджень та тестувань фірмою «Віаком» (м. Київ). Пропускна здатність реалізованого декодеру залежить від конкретного типу коду, який подається на вхід, а також налаштувань декодеру (зокрема, щодо кількості ітерацій). Для кодів, які відносяться до групи з довжиною 64 ((64,57), (63,56), (58,51), (46,39)) пропускна здатність становила понад 200 Мбіт/с при 10 повноцінних ітераціях (відповідно, при кращій якості сигналу та меншому рівні шуму даний показник збільшується).

1. Berrou C. Near Shannon limit error-correcting coding and decoding : Turbo-codes / C. Berrou, A. Glavieux, P. Thitimajshima – IEEE ICC'93, 1993. – pp. 1064–1071.
2. Goubier T. Fine Grain Parallel Decoding of Turbo Product Codes: Algorithm and Architecture / T. Goubier, C. Dezan, B. Pottier, C. Jego – 2008 5th International Symposium on Turbo Codes and Related Topics, 2008. DOI: 10.1109/TURBOCODING.2008.4658678.
3. Krainyk Y. Low-complexity high-speed soft-hard decoding for turbo-product codes / Y. Krainyk, V. Perov, M. Musiyenko – 2017 IEEE 37th International Conference on Electronics and Nanotechnology (ELNANO), 2017. – Kyiv, 2017. – pp. 471-474.
4. Krainyk Y. Hardware-Oriented Turbo-Product Codes Decoder Architecture / Krainyk Y., Perov V., Musiyenko M., Davydenko Ye. // Conference Proceedings of IEEE IDAACS-2017. - Bucharest, 2017. -pp. 151-154.
5. Zhou L. Flexible and high-efficiency turbo product code decoder design / Zhou L., Liu H., Zhang B. // IEICE Electronics Express, Vol. 9, No. 12. – 2012. – pp. 1044-1050. DOI: 10.1587/elex.9.1044
6. Megha S. Iterative Decoding Algorithm for Turbo Product Codes / Megha M.S. / International Journal of Innovative Research and Advanced Engineering (IJIRAE), Vol. 1, Issue 2. – April 2014. – pp.151-154.
7. Han J. High Speed Max-Log-MAP Turbo SISO Decoder Implementation Using Branch Metric Normalization / Han J., Erdogan E., Arslan T. / Proceedings of the IEEE Computer Society Annual Symposium on VLSI. – 2005. DOI: 10.1109/ISVLSI.2005.37
8. Mathana J. FPGA Implementation of High Speed Architecture for Max Log Map Turbo SISO Decoder / Mathana J., Rangarajan Dr. / International Journal of Recent Trends in Engineering, Vol. 2, No. 6. – 2009. – pp. 142-146.

УДК 65.011.56

Кузьмич О.І. к.т.н., Маркіна Л.М. асистент, Якимчук Н.М. асистент.

Луцький НТУ

МОДЕЛЮВАННЯ ТА РОЗРОБКА СИСТЕМИ КЕРУВАННЯ ПРОЦЕСОМ ЗМІШУВАННЯ НА БАЗІ ПРОЦЕСУ ВОДНО-ТЕПЛОВОЇ ОБРОБКИ ПРИ ВИРОБНИЦТВІ СПИРТУ

Кузьмич О.І., Маркіна Л.М., Якимчук Н.М. Моделювання та розробка систему керування процесом змішування на базі процесу водно-теплової обробки при виробництві спирту. У статті проведено моделювання та розроблено систему керування процесом змішування на базі процесу водно-теплової обробки при виробництві спирту.

Ключові слова: система керування, математичне моделювання, водно-теплова обробка, ПІД регулятор, лінеаризація, стійкість.

Кузьмич А.И., Маркина Л., Якимчук Н.М. Моделирование и разработка системы управления процессом смешивания на базе процесса водно-тепловой обработки при производстве спирта. В статье проведено моделирование и разработана система управления процессом смешивания на базе процесса водно-тепловой обработки при производстве спирта.

Ключевые слова: система управления, математическое моделирование, водно-тепловая обработка, ПИД регулятор, линейаризация, устойчивость.

Kuzmich O.I., Markina L.M., Yakymchuk N.M. Modeling and development of a mixing process control system based on the process of water and heat treatment in the production of alcohol. In the article the modeling and the system of control of mixing process on the basis of the process of water-heat treatment at the production of alcohol are developed.

Key words: control system, mathematical modeling, water-heat treatment, PID regulator, linearization, stability.

Постановка проблеми. Спиртова галузь є однією з найприбутковіших галузей української економіки, вона відіграє важливу роль у ефективному функціонуванні інших галузей: фармацевтичної, харчової, агропромислового та паливно-енергетичного комплексів.

Спиртова промисловість України на даний час являє собою одну із найбільших, науково і технічно розвинутих галузей господарства. Теперішня потужність заводів була досягнута в середині 90-х років і досі поступово зростає за рахунок реконструкцій та введення у дію нових відділень, при цьому систематично замінюється застаріла апаратура.

Незадовільна робота обладнання та несвоечасний контроль і регулювання основних параметрів технологічного процесу часто призводить до підвищених втрат, зниженню якості кінцевого продукту, зниженню продуктивності установок і значної перевитрати енергії й вторинних ресурсів. Все це істотно впливає на економічні показники виробництва спирту і обумовлює зайву витрату матеріальних цінностей, у першу чергу таких важливих продуктів, як зерно, картопля, м'яса та ін.

Сьогодні вітчизняна спиртова промисловість за рік може випускати більше 60 млн. дал спирту. Ставиться завдання покращити якісні показники спирту та горілки, знизити їх собівартість та підвищити ефективність переробки.

Аналіз останніх досліджень. Технологія спирту як наука пройшла довгий шлях розвитку, перш ніж досягла високого сучасного науково-технічного рівня, у створенні і вдосконаленні брали участь видатні вчені, інженери багатьох країн. Значний внесок у розвиток теорії і практики спиртового виробництва внесли такі вчені, як В. О. Маринченко, В.М. Швець, О. М. Плановського, М. М. Ліпатова, О. В. Ликова, А. С. Гінзбурга, С. М. Гребенюка, М. О. Гришина, Г. А. Аксельруда, М. І. Беляєва та ін. Досліджувана нами тематика висвітлюється в роботах багатьох науковців, але з точки зору технології, а саме розробка математичної моделі та системи керування дозволить нам оптимізувати даний технологічний процес.

Формулювання мети дослідження. Розробка ефективної системи регулювання процесом водно-теплової обробки.

Виклад основного матеріалу. Підготовка крохмалевмісної сировини до зброджування в етиловий спирт складається з таких технологічних стадій: подрібнення сировини, змішування помелу з водою (приготування замісу), попередній підігрів замісу, водно-теплова обробка сировини, оцукрювання розвареної маси.

В якості крохмалевмісної сировини, що переробляється на спирт, використовується зерно або картопля. Цю сировину змішують з водою і подають на розварювання, причому концентрація

такого замісу визначається вмістом в ній крохмалю. Для забезпечення гомогеності замісу його перемішують пропелерною мішалкою. Мета розварювання замісів сировини - вивільнити крохмаль з рослинних клітин та перевести крохмаль у розчинний стан. У процесі розварювання проходить також стерилізація замісів, що важливо в подальших технологічних процесах оцукрювання і зброджування, тому ставиться завдання своєчасного контролю та регулювання параметрів даного технологічного процесу. Для досягнення ефективного автоматизованого регулювання спочатку необхідно провести моделювання. Кожний технологічний процес потребує індивідуального підходу, але є загальні, універсальні рішення оптимізації для багатьох процесів, які можуть бути сформульовані.

Моделюванням називається метод дослідження існуючого чи створюваного об'єкта, коли замість об'єкта (оригіналу) вивчається модель (другий об'єкт, що замінює оригінал), а результати кількісно переносяться на оригінал.

Метод моделювання дозволяє уникнути великих помилок під час реалізації виробництва і прискорює його освоєння. Багато процесів неможливо вивчати у промислових апаратах унаслідок неможливості довільної зміни режиму їх роботи, труднощів роздільного вивчення спільно діючих чинників, можливого псування великої кількості продуктів. На сьогоднішній день існує багато методів, які дозволяють дослідити та змоделювати процес, який досліджується. Кожен з методів має ряд недоліків та переваг.

Наявність "сильних" і "слабких" сторін аналітичного й експериментального методів складання математичного опису привело до розробки комбінованого експериментально-аналітичного методу. Сутність його полягає в аналітичному складанні рівнянь опису, проведенні експериментальних досліджень і отриманні по їхніх результатах параметрів рівнянь. Для дослідження даного технологічного процесу, а також і для подальшого моделювання найбільш доцільно використати даний метод для моделювання водно-теплової обробки замісу.

Процеси перемішування широко використовуються в харчовій промисловості для утворення однорідних продуктів з визначеними властивостями, а також для інтенсифікації технологічних процесів. Перемішування здійснюють різними способами в різних апаратах з мішалками, а також у газових і рідинних потоках.

Згідно існуючих гіпотез про механізм перемішування математичний опис представляється у вигляді залежностей між параметрами, що характеризують механічний рух часток і факторами, що викликають їхнє переміщення.

Розглянемо декілька способів перемішування, які найчастіше зустрічаються в харчовій промисловості: регулювання рівня шляхом зміни витрати суміші, регулювання за допомогою регулятора співвідношення, регулювання барботажних змішувачів.

Провівши аналіз існуючих систем управління процесом змішування двох складових, було встановлено, що найбільше для дослідження, моделювання даного технологічного процесу підходять наступні математичні залежності, які будуть розглянуті нижче.

Розглянемо систему управління процесом змішування рідин. Схема процесу зображена на рис. 1. Ємність наповнюється з допомогою двох потоків, що мають змінні миттєві витрати $F_1(t)$, $F_2(t)$. Обидва вхідних потоки містять розчинну рідину з постійними концентраціями c_1, c_2 .

Вихідний потік має масову швидкість витікання $F(t)$. Передбачається, що вміст бака перемішується так, що концентрація вихідного потоку дорівнює концентрації $c(t)$ в баку.

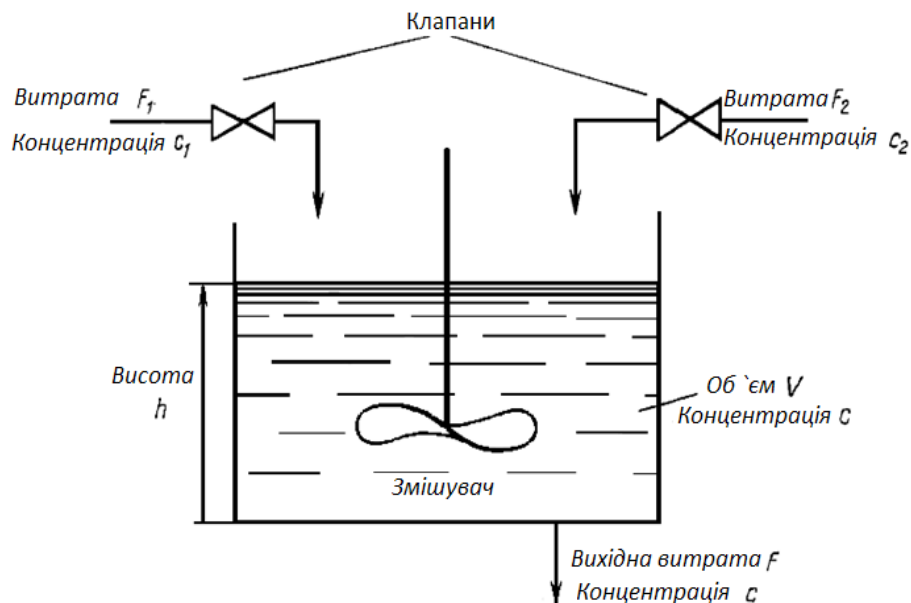


Рис. 1. Змішувальна ємність

Рівняння балансу мас для ємності мають вигляд:

$$\begin{aligned} \frac{dV(t)}{dt} &= F_1(t) + F_2(t) - F(t), \\ \frac{d}{dt}[c(t)V(t)] &= c_1 F_1(t) + c_2 F_2(t) - c(t)F(t), \end{aligned} \quad (1)$$

де $V(t)$ – об'єм рідини у баку. Миттєва витрата вихідного потоку $F(t)$ залежить від висоти $h(t)$ наступним чином :

$$F(t) = k\sqrt{h(t)}, \quad (3)$$

де k – експериментальна константа. Якщо ємність має постійну площу поперечного перерізу S , то можна записати:

$$F(t) = k\sqrt{\frac{V(t)}{S}}; \quad (4)$$

тоді рівняння балансу мас приймуть вигляд:

$$\begin{aligned} \frac{dV(t)}{dt} &= F_1(t) + F_2(t) - k\sqrt{\frac{V(t)}{S}}, \\ \frac{d}{dt}[c(t)V(t)] &= c_1 F_1(t) + c_2 F_2(t) - c(t)k\sqrt{\frac{V(t)}{S}}. \end{aligned} \quad (5)$$

Розглянемо спочатку випадок стаціонарного стану, коли всі величини є постійними: F_{10} , F_{20} , та F_0 – витрати, V_0 – об'єм, і c_0 – концентрація в баку. Тоді мають місце такі співвідношення:

$$0 = F_{10} + F_{20} - F_0, \quad (7)$$

$$0 = c_1 F_{10} + c_2 F_{20} - c_0 F_0, \quad (8)$$

$$F_0 = k\sqrt{\frac{V_0}{S}}. \quad (9)$$

При заданих F_{10} і F_{20} ці рівняння можуть бути вирішені відносно F_0 , V_0 та c_0 . Припустимо тепер, що виникли невеликі відхилення від встановленого стану. Запишемо:

$$F_1(t) = F_{10} + \mu_1(t),$$

$$F_2(t) = F_{20} + \mu_2(t), \quad (10)$$

$$V(t) = V_0 + \xi_1(t),$$

$$c(t) = c_0 + \xi_2(t),$$

Де μ_1 і μ_2 розглядаються як вхідні змінні, а ξ_1 і ξ_2 – змінні стану. У припущенні, що зазначені чотири параметри є малими, лінеаризація (5) та (6) приводить до рівнянь:

$$\dot{\xi}_1(t) = \mu_1(t) + \mu_2(t) - \frac{k}{2V_0} \sqrt{\frac{V_0}{S}} \xi_1(t), \quad (11)$$

$$\xi_2(t)V_0 + c_0 \dot{\xi}_1(t) = c_1 \mu_1(t) + c_2 \mu_2(t) - c_0 \frac{k}{2V_0} \sqrt{\frac{V_0}{S}} \xi_1(t) - k \sqrt{\frac{V_0}{S}} \xi_2(t). \quad (12)$$

Підставляючи (9) в ці рівняння, отримаємо:

$$\dot{\xi}_1(t) = \mu_1(t) + \mu_2(t) - \frac{1}{2} \frac{F_0}{V_0} \xi_1(t), \quad (13)$$

$$\xi_2(t)V_0 + c_0 \dot{\xi}_1(t) = c_1 \mu_1(t) + c_2 \mu_2(t) - \frac{1}{2} c_0 \frac{F_0}{V_0} \xi_1(t) - F_0 \xi_2(t). \quad (14)$$

Введемо параметр:

$$\frac{V_0}{F_0} = 0, \quad (15)$$

що називається часом заповнення ємності. Виключення $\dot{\xi}_1$ із (14) призводить до лінеаризованого диференціального рівняння стану:

$$\dot{x}(t) = \begin{pmatrix} -\frac{1}{2\theta} & 0 \\ 0 & -\frac{1}{\theta} \end{pmatrix} x(t) + \begin{pmatrix} \frac{1}{V_0} & \frac{1}{V_0} \\ \frac{c_1 - c_0}{V_0} & \frac{c_2 - c_0}{V_0} \end{pmatrix} u(t), \quad (16)$$

де $x(t) = \text{col}[\xi_1(t), \xi_2(t)]$ та $u(t) = \text{col}[\mu_1(t), \mu_2(t)]$;

Якщо визначити вихідні змінні у вигляді:

$$\eta_1(t) = F(t) - F_0 \approx \frac{1}{2} \frac{F_0}{V_0} \xi_1(t) = \frac{1}{2\theta} \xi_1(t),$$

$$\eta_2(t) = c(t) - c_0 = \xi_2(t), \quad (17)$$

то можна доповнити рівняння (16) лінеаризованим рівнянням вихідної змінної:

$$y(t) = \begin{pmatrix} \frac{1}{2\theta} & 0 \\ 0 & 1 \end{pmatrix} x(t), \quad (18)$$

де $y(t) = \text{col}[\eta_1(t), \eta_2(t)]$. Далі підставляємо дані, які можна отримати експериментальним способом.

В результаті лінеаризована система рівнянь прийме вигляд:

$$\dot{x}(t) = \begin{pmatrix} -0,01 & 0 \\ 0 & -0,02 \end{pmatrix} x(t) + \begin{pmatrix} 1 & 1 \\ -0,25 & 0,75 \end{pmatrix} u(t),$$

$$y(t) = \begin{pmatrix} 0,01 & 0 \\ 0 & 1 \end{pmatrix} x(t). \quad (19)$$

При $u(t) \equiv 0$ лінеаризована система описується рівнянням

$$\dot{x}(t) = \begin{pmatrix} -\frac{1}{2\theta} & 0 \\ 0 & -\frac{1}{\theta} \end{pmatrix} x(t), \quad (20)$$

яке має рішення

$$\xi_1(t) = e^{-t/2\theta} \xi_1(0), \quad t \geq 0,$$

$$\xi_2(t) = e^{-t/\theta} \xi_2(0), \quad t \geq 0. \quad (21)$$

Очевидно, $\xi_1(t)$ і $\xi_2(t)$ завжди наближаються до нульового значення при зростанні t , так як $\theta > 0$. В результаті, лінеаризована система асимптотично стійка. Більш того, оскільки збіжність до положення рівноваги є експоненціальною, система експоненціально стійка.

Із проведених розрахунків видно, що якщо лінеаризована система асимптотично стійка, то положення рівноваги, відносно якого була проведена лінеаризація, є асимптотично стійким, але не обов'язково асимптотично стійким в цілому.

З фізичних розумінь, однак, слід очікувати, що в даному випадку система є асимптотично стійкою в цілому.

Для подальших розрахунків припустимо, що потрібно встановити такий режим роботи системи, при якому підтримуються постійна витрата $F(t)$ і постійна концентрація $c(t)$, які є основними параметрами для ефективного керування технологічним процесом водно-теплової обробки. Один з методів досягнення цієї мети полягає в регулюванні витрати F шляхом зміни витрати основного потоку F_1 і регулювання концентрації з допомогою зміни витрати іншого потоку F_2 .

Обираємо наступні закони зміни вхідних змінних:

$$\begin{aligned} \mu_1(t) &= -k_1 n_1(t), \\ \mu_2(t) &= -k_2 n_2(t). \end{aligned} \quad (22)$$

Цей вираз показує, що для додатних значень k_1 і k_2 замкнута система є стійкою. Виберемо наступні значення для коефіцієнтів підсилення: $k_1 = 10$ і $k_2 = 0,1$.

Припустимо, що обидва потоки, надходять в змішувальну ємність, мають рівні концентрації $c_1 = c_2 = \bar{c}$. Тоді установлена концентрація в баку c_0 також дорівнює $\bar{c}$, а лінеаризоване диференціальне рівняння стану має вигляд

$$\dot{x}(t) = \begin{pmatrix} -\frac{1}{20} & 0 \\ 0 & -\frac{1}{0} \end{pmatrix} x(t) + \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} u(t). \quad (23)$$

З цього рівняння видно, що приріст концентрації, яка є другою компонентою стану, не може управлятися за допомогою вектора вхідної змінної, компонентами якого є збільшення вхідних потоків. Фізично це також ясно, так як за припущенням вхідні потоки мають рівні концентрації.

Попередня математична модель є відносно проста, так як об'єкт має скалярні вхідні і керовані змінні. Багатовимірні завдання управління, в яких об'єкт має кілька вхідних і декілька керованих змінних, набагато складніше, тому наступні дослідження і розрахунки найбільше відтворюватимуть наш технологічний процес та найбільш ефективно проводитимуть регулювання. Далі розглянемо багатовимірну систему на прикладі змішувальної ємності. Ємність має два підведення живлення, а відповідні потоки можуть регулюватися клапанами. Концентрація розчинної речовини в кожному з потоків постійна і не може бути змінена. Ємність має один вихідний отвір і завдання управління полягає в проектуванні установки, яка б автоматично регулювала витрати через клапани живлення з тим, щоб підтримувати витрату і концентрацію вихідного потоку постійними відповідно до заданих еталонних значень.

Це типова задача регулювання. Компонентами вхідної змінної є витрати вхідних потоків. Компонентами керованої змінної є вихідна витрата і концентрація вихідного потоку. Задана робоча точка також має дві компоненти: бажана вихідна витрата і бажана вихідна концентрація.

Можуть мати місце такі збурення: флуктуації вхідних концентрацій, флуктуації вхідних витрат внаслідок флуктуацій тисків перед клапанами, втрат рідини через витоки.

Для правильного управління системою повинні вимірюватися і вихідна витрата, і концентрація; тоді вони з'являться компонентами спостерегаючої змінної. Замкнений регулятор використовує як ці зміни, так і значення заданих рівнів витрати і концентрації для формування пневматичного або електричного сигналу, який регулює клапани. Далі досліджуємо процес в динаміці, з використанням ПІД регулятора.

Закон управління протесовано для різних операційних точок функціонування системи. Моделювання динаміки системи виконано в програмному середовищі Matlab / Simulink (рис. 2).

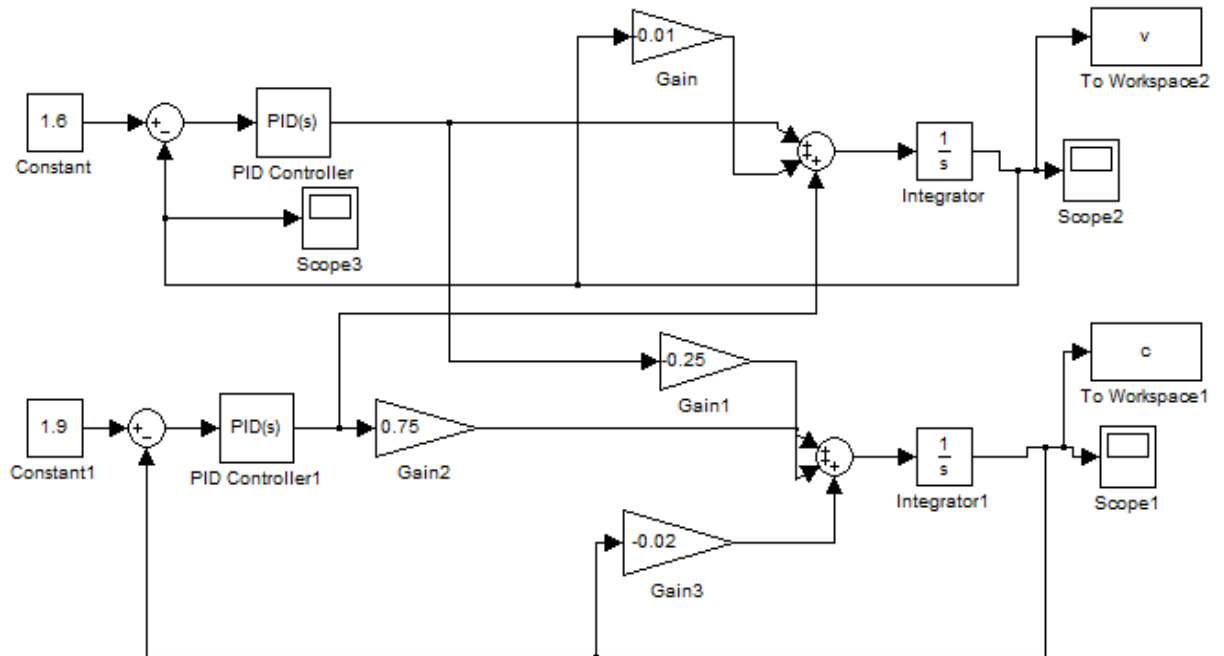


Рис. 2. Моделювання динаміки системи керування процесом водно-теплової обробки.

Після проведеної серії експериментів для кількох операційних точок ми отримали наступні результати (рис.3).

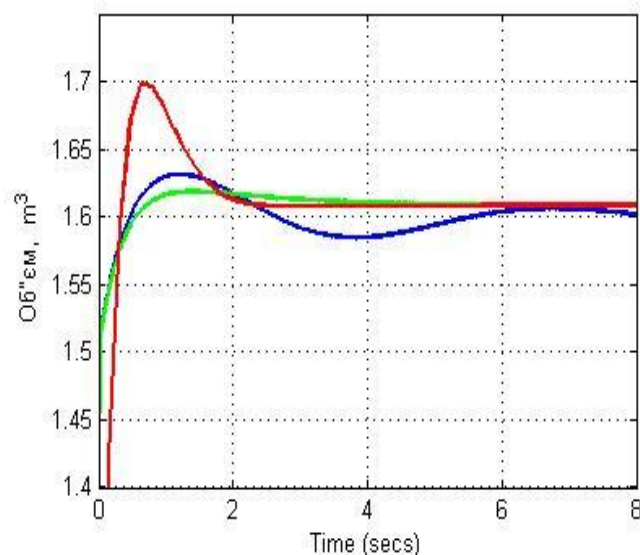


Рис. 3. Реалізація ПІД контролера.

Висновки. Проаналізувавши результати відкликів контролера, можна зробити висновок, що наявність усіх трьох компонентів ПІД регулятора дає найякісніший відклик системи стосовно швидкодії, стабільності роботи, відсутності значних відхилень і збурень.

Перспективи подальших досліджень. Розробка імітаційної моделі для даного технологічного процесу.

1. Голубятников В.А., Шувалов В.В. Автоматизация производственных процессов в химической промышленности: Учебн. Для техникумов. – 2-е изд., перераб. и доп. – М.: Химия, 1985. – 352с., ил.
2. Домарецький В.А., Шиян П.Л., Швець В.М., Циганков П.С., Жолнер І.Д., Маринченко В.А. Технологія спирту. – Вінниця: Поділля-2000, 2003 – 496 с.
3. Квакуернаак Х., Сиван Р. Линейные оптимальные системы управления. М.: Мир, 1977. — 656 с.
4. Клавдиев А.А. Теория автоматического управления в примерах и задачах Ч.І.: Учеб. Пособие.- СПб:СЗТУ, 2005. –74с.
5. Лазарева Т. Я., Мартемьянов Ю. Ф. Основы теории автоматического управления: Учебное пособие. 2-е изд., перераб. и доп. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2004. –352 с.
6. Маринченко В.О., Домарецький В.А., Шиян П.Л., Швець В.М., Циганков П.С., Жолнер І.Д. Технологія спирту. /Под ред. проф. Маринченка В.О. - Вінниця: “ Поділля-2000” , 2003. - 496 с.
7. Потапов В.О. Моделювання технологічних процесів харчових виробництв. Навчальний посібник: - Х.: ХДУХТ, 2008 – 148 с. іл. – 23.
8. Черевко О.І., Поперечний А. М. Процеси і апарати харчових виробництв: підручник / Черевко О. І., Поперечний А. М. — 2-е видання, доп. та випр. — Х.: Світ Книг, 2014. - 495 с.
9. Яровенко В.Л., Маринченко В.А., Смирнов В.А. Под ред проф. Яровенко В.Л. – М.: Колос, 2002.
10. Дорош А.К. Производство спиртных напитков. Сырье, аппараты, технологии получения спирта и водки. К.: Либідь, 1995. – 272с.

УДК 519.876.5

В.В. Лишук, Й.Р. Селепина, Л.В. Ящинський, С.М. Костючко
Луцький національний технічний університет

ЗАСТОСУВАННЯ МЕТОДУ ЕЙЛЕРА В ЗАДАЧАХ ДИНАМІКИ

В.В. Лишук, Й.Р. Селепина, Л.В. Ящинський, С.М. Костючко. Застосування методу Ейлера в задачах динаміки. У статті показано застосування явного методу Ейлера на прикладі математичної моделі руху повітряної кулі, що описується нелінійним диференціальним рівнянням. Для опису та аналізу фізичних процесів застосовано мову програмування *FORTRAN* з графічним редактором *GRAPHER*.

Ключові слова: повітряна куля, алгебраїчні та диференціальні рівняння, інтегрування рівнянь.

В.В. Лышук, И.Р. Селепина, Л.В. Яшинский, С.М. Костючко. Применение метода Эйлера в задачах динамики. В статье показано применение явного метода Эйлера на примере математической модели движения воздушного шара, которая описывается нелинейными дифференциальными уравнениями. Для описания и анализа физических процессов применен язык программирования *FORTRAN* с графическим редактором *GRAPHER*.

Ключевые слова: воздушный шар, алгебраические и дифференциальные уравнения, интегрирование уравнений.

W.W. Lyshuk, Y.R. Selepyna, L.W. Jashchynskyy, S.M. Kostyutcho. Application of the Euler method in dynamics problems. In the article the application of the explicit Euler method is shown on the example of a mathematical model of the balloon motion, which is described by nonlinear differential equations. The *FORTRAN* programming language with *GRAPHER* graphical editor is used for describing and analyzing physical processes.

Keywords: balloon, algebraic and differential equations, integration of equations.

Постановка проблеми. На сьогодні часто доводиться стикатися із задачами, розв'язок яких класичними методами неможливий. Це фізичні задачі, що описуються лінійними чи нелінійними диференціальними рівняннями зі звичайними або частинними похідними. Як відомо, розв'язком диференціального рівняння є певна функція або множина точок, що в залежності від кроку дискретизації і кінцевого часу інтегрування може сягати десятків-сотень тисяч точок, які необхідно розрахувати. Розв'язати таке рівняння вручну неможливо через величезний об'єм роботи. Тут слід застосувати інтегрування такого рівняння числовими методами [3, 4].

Якщо рівняння зі звичайними похідними розв'язуються явними чи неявними методами (Ейлера, Рунге-Кутта), то до рівнянь з частинними похідними слід застосовувати просторову дискретизацію за методами скінченних різниць або скінченних елементів. Це реалізується з допомогою персональних комп'ютерів з використанням відповідних математичних методів та програмування [1].

Аналіз досліджень. Класична математика пройшла довгий етап розвитку і досягла величезних успіхів у інтелектуальній культурі людства. Широке застосування ЕОМ в математичному моделюванні, розроблена теорія і значні практичні результати дають змогу говорити про обчислювальний експеримент, як про нову технологію і методологію наукових і практичних досліджень.

Відзначимо деякі переваги обчислювального експерименту в порівнянні з натурним. Обчислювальний експеримент, як правило, дешевший за фізичний. В цей експеримент можна легко вносити певні зміни. Його можна повторити ще раз, якщо це необхідно, і перервати в будь-який момент. В ході такого експерименту можна змоделювати умови, які не можна створити в лабораторії [3].

У ряді випадків проведення натурального експерименту затруднено, а іноді й неможливо. Часто проведення такого експерименту пов'язане з можливими непередбачуваними наслідками або з небезпекою для життя чи здоров'я людей. В цих випадках обчислювальний експеримент може стати основним засобом дослідження. З його допомогою виявляється можливим прогнозувати властивості нових, ще не створених конструкцій і матеріалів на стадії їх проектування, а також моделювати складні фізичні процеси. У той же час потрібно пам'ятати, що застосування результатів обчислювального експерименту обмежене рамками прийнятої математичної моделі. На відміну від натурних досліджень, обчислювальний експеримент дає змогу накопичувати отримані результати [4].

Отож, поява ЕОМ дала потужний поштовх ще більш широкому впровадженню чисельних методів в практику наукових і технічних розрахунків. Швидкість виконання обчислювальних операцій зросла в мільйони разів, що дало змогу вирішувати широкий клас математичних задач, які до цього практично не розв'язувались.

Виклад основного матеріалу й обґрунтування отриманих результатів. У нашому столітті літаків і ракет, для яких доступні будь-які висоти над поверхнею Землі, повітряні кулі, громіздкі, ненадійні і некеровані, вже відійшли в минуле, хоча колись саме вони дали людині можливість піднятися в повітря. Втім, у деяких випадках повітряні кулі дуже зручні і часто використовуються. Наприклад, з аеростата зручно навчати стрибків з парашутом, а метеорологи можуть досліджувати тиск, температуру і повітряні потоки в атмосфері за допомогою куль-зондів.

Побудуємо математичну модель задачі, в якій потрібно визначити швидкість піднімання кулі, максимальну висоту підйому у функції часу. При цьому задається деяка модель атмосфери, тобто закон зміни тиску, густини або температури в залежності від висоти.

По суті, ця задача полягає у знаходженні умови рівноваги кулі, на яку діє сила тяжіння Землі, виштовхувальна (архімедова) сила і сила опору середовища. Якщо виштовхуюча сила більша за силу тяжіння (різниця цих сил називають підйомної силою), куля піднімається вгору. Але в міру підйому зменшується густина повітря, а, отже, зменшуються виштовхувальна сила і сила опору.

При збільшенні радіуса кулі виштовхувальна сила зростатиме пропорційно кубу радіуса, а вага оболонки – пропорційно квадрату радіусу. Отже, виштовхувальна сила зростатиме швидше і, починаючи з певного радіусу, стане більшою, ніж вага оболонки. Тоді куля почне підніматися вгору.

Розглянемо задачу динаміки тіла змінної маси, а саме піднімання повітряної кулі, що тягне за собою складений канат. Маса кулі змінюється за законом:

$$m = m_0 + \frac{\gamma h}{g} = \frac{G_0 + \gamma h}{g}, \quad (1)$$

де m_0 – маса повітряної кулі без канату, кг, γ – питома погонна вага канату, Н/м, h – висота підйому кулі, $g = 9,81 \frac{\text{м}}{\text{с}^2}$ – прискорення вільного падіння, G_0 – вага кулі, Н.

Об'єм кулі W , площу поперечного перерізу S будемо вважати незмінними, а довжину і міцність канату – достатніми.

Густина повітря, як відомо, зі збільшенням висоти зменшується. Опишемо цю зміну експоненціальним законом:

$$\rho = \rho_0 \cdot e^{-\alpha h}, \quad (2)$$

де ρ_0 – густина повітря на поверхні землі (на рівні моря), кг/м^3 , α – коефіцієнт зменшення густини, м^{-1} .

Як відомо, на кулю діють такі сили: підйомальна сила, сила тяжіння і сила опору повітря.

Для побудови моделі опишемо ці сили.

Архімедова сила:

$$P = \rho g W. \quad (3)$$

Сила тяжіння:

$$G = G_0 + \gamma h, \quad (4)$$

де G_0 – вага кулі, Н:

$$G_0 = S' z = 4\pi r^2 z. \quad (5)$$

Тут S' – площа поверхні шару, м^2 , z – густина матеріалу кулі, Н/м^2 , r – радіус кулі, м.

Сила опору повітря:

$$R = \frac{k S \rho V^2}{2}. \quad (6)$$

Тут k – коефіцієнт лобового опору, V – швидкість кулі, м/с , $S = \pi r^2$ – площа поперечного перерізу кулі, м^2 .

Складемо диференціальне рівняння у формі Мещерського [1] для вертикального руху кулі:

$$m \frac{d^2 h}{dt^2} = -\frac{dh}{dt} \cdot \frac{dm}{dt} - (G_0 + \gamma h) + \rho_0 e^{-\alpha h} \left(gW - 0,5kS \left(\frac{dh}{dt} \right)^2 \right). \quad (7)$$

Оскільки G_0, γ – постійні, то:

$$\frac{dm}{dt} = \frac{d((G_0 + \gamma h)/g)}{dt} = \frac{\gamma}{g} \cdot \frac{dh}{dt}. \quad (8)$$

Поділивши рівняння (7) на (1) і врахувавши, що $\frac{d^2 h}{dt^2} = \frac{dV}{dt}$ або $\frac{dh}{dt} = V$, після елементарних перетворень отримаємо:

$$\left. \begin{aligned} \frac{dV}{dt} &= \frac{-g + [\rho_0 e^{-\alpha h} g^2 W - (\gamma + b\rho_0 e^{-\alpha h} G_0) V |V|]}{G_0 + \gamma h}, \\ \frac{dh}{dt} &= V, \end{aligned} \right\} \quad (9)$$

У системі рівнянь (9) врахована зміна знаку сили опору повітря при опусканні кулі, а множник b називається балістичним коефіцієнтом:

$$b = \frac{kS}{2m_0} = \frac{kgS}{2G_0}. \quad (10)$$

Сумісному чисельному інтегруванню підлягає система нелінійних рівнянь (9). Умовою кінця інтегрування можна вибрати досягнення максимальної висоти або отримання першого від'ємного значення швидкості V .

Для практичного користування запропонованою моделлю необхідно знати густину атмосфери ρ_0 , коефіцієнт зменшення густини повітря α , радіус кулі r , густину матеріалу кулі z , погонну вагу канату γ , коефіцієнт лобового опору k . Боковим вітром нехтуємо.

Для моделі приймаємо такі параметри:
 $g = 9,81 \text{ м/с}^2$; $\pi = 3,14$; $\rho_0 = 1,225 \text{ кг/м}^3$; $\alpha = 0,0002 \text{ м}^{-1}$;
 $k = 0,005$; $r = 1,3 \text{ м}$; $z = 0,7 \text{ Н/м}^2$; $\gamma = 0,15 \text{ Н/м}$; $G_0 = 10 \text{ Н}$.

Інтегрування системи нелінійних диференціальних рівнянь (9) здійснено явним методом Ейлера. Крок часової дискретизації $dt = 0,01 \text{ с}$. Для розрахунку перехідних процесів використано мову програмування *Visual FORTRAN* з графічним пакетом *GRAPHER*.

Метод Ейлера ґрунтується на заміні шуканої функції многочленом І-го степеня, тобто на лінійній екстраполяції. Розглянемо диференціальні рівняння нашої задачі з нульовими початковими умовами (куля нерухома).

$$\frac{dx}{dt} = f(x, t), \quad x = V, h; \quad V(t_0) = V^{(0)}, \quad h(t_0) = h^{(0)}. \quad (11)$$

Розглянемо інтервал $[a, b]$ і розіб'ємо його на m відрізків вузлами t_k з кроком Δt . Точний розв'язок рівняння (11) у точці t_{k+1} на будь-якому з окремих інтервалів $[t_k, t_{k+1}]$ $0 \leq k \leq m-1$ можна представити у вигляді ряду Тейлора з центром у точці x_k .

Візьмемо до уваги, що функція $f(x, t) \in n+1$ разів диференційована за обома аргументами [4], тобто:

$$x(t_{k+1}) = x(t_k) + \Delta t \cdot x'(t_k) + \frac{\Delta t^2}{2!} \cdot x''(t_k) + \frac{\Delta t^3}{3!} \cdot x'''(t_k) + \dots + \delta(\Delta t^{(n+1)}). \quad (12)$$

Похідні у (12) можна обчислити згідно (11) таким чином:

$$\begin{aligned} x'(t) &= f(x, t); & x''(t) &= \frac{\partial f}{\partial t}(x, t) + f \cdot \frac{\partial f}{\partial x}(x, t); \\ x'''(t) &= \frac{\partial}{\partial t} \left[\frac{\partial f}{\partial t} + f \cdot \frac{\partial f}{\partial x} \right] + \frac{\partial}{\partial x} \left[\frac{\partial f}{\partial t} + f \cdot \frac{\partial f}{\partial x} \right] \cdot f; & x^{(n)}(t) &= \frac{\partial}{\partial t} [x^{(n-1)}(t)] + \frac{\partial}{\partial x} [x^{(n-1)}(t)] \cdot f. \end{aligned} \quad (13)$$

Якщо прийняти $x = x(t)$ за точний розв'язок (11), а потім підставити $t = t_k$ і відкинути залишковий член у (12), то отримаємо таке дискретне рівняння:

$$\begin{aligned} x_{k+1}(t) &= x_k + \Delta t \cdot f(x_k, t_k) + \frac{\Delta t^2}{2!} \left[\frac{\partial f}{\partial t}(x_k, t_k) + \frac{\partial f}{\partial x}(x_k, t_k) \cdot f(x_k, t_k) \right] + \dots + \\ &+ \frac{\Delta t^n}{n!} \left[\frac{\partial^{(n-1)}}{\partial t^{(n-1)}} \cdot f(x_k, t_k) + \dots + \frac{\partial^{(n-1)}}{\partial t^{(n-1)}} \cdot f(x_k, t_k) \cdot f^{(n-1)} \right], \end{aligned} \quad (14)$$

де $0 \leq i \leq n-1$, $x(t_0) = x^{(0)}$, $x = V$, h задано в (11).

Рівняння (12) є двоточною явною різницевою схемою, що дає змогу обчислити послідовно всі значення, починаючи з x_1 і закінчуючи x_m . Похибка апроксимації при цьому дорівнюватиме значенню відкинутого доданка у (12) при $\Delta t \rightarrow 0$.

Якщо прийняти $n = 1$, то отримаємо шуканий явний метод Ейлера, причому точність апроксимації буде $\delta(\Delta t^2)$:

$$x_{k+1} = x_k + \Delta t \cdot f(x_k, t_k). \quad (15)$$

На рис.1, 2 показано розроблену програму у середовищі *FORTRAN* та після компіляції програми біжучий файл результатів *.exe* на основі якого формується масив змінних, що записується і зберігається у файл з розширенням *.dat*.

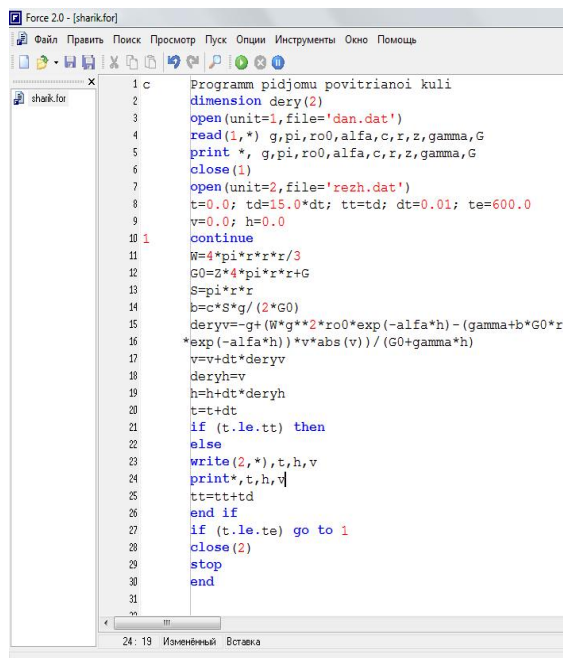


Рис.1. Вікно розробленої програми у середовищі *FORTRAN FORCE 2.0*.



Рис.2. Вікно файлу результатів *.exe*.

Перехідні процеси швидкості кулі та її висоти в функції часу показано на рис.3, 4, а на рис.5 блок-схему програми.

Просимульовано підйом кулі. На рис.3, 4 видно, що максимальна швидкість кулі становить $V = 21$ м/с при часі $t = 1,95$ сек. Після цього швидкість зменшується і в момент часу $t = 47$ сек швидкість дорівнює нулю, а це означає, що висота піднімання в нашому випадку є максима-

льною і становить $h = 542$ м. Далі швидкість кулі стає від'ємною, досягає свого максимального від'ємного значення $V = -3$ м/с, що означає опускання повітряної кулі. Далі спостерігаємо згасаючі коливання з періодом 55 сек.

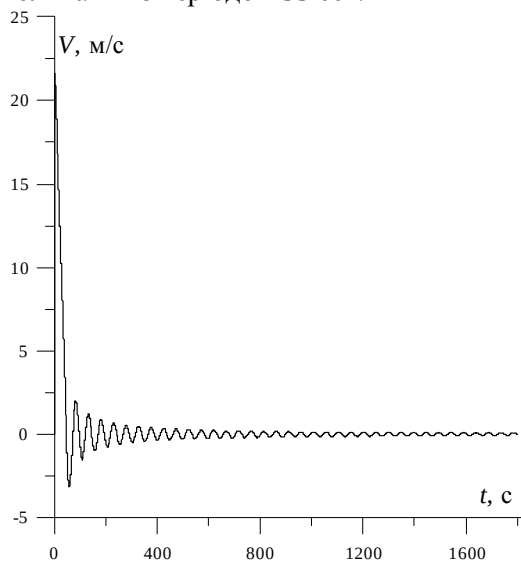


Рис.3. Графік швидкості під'йому кулі.

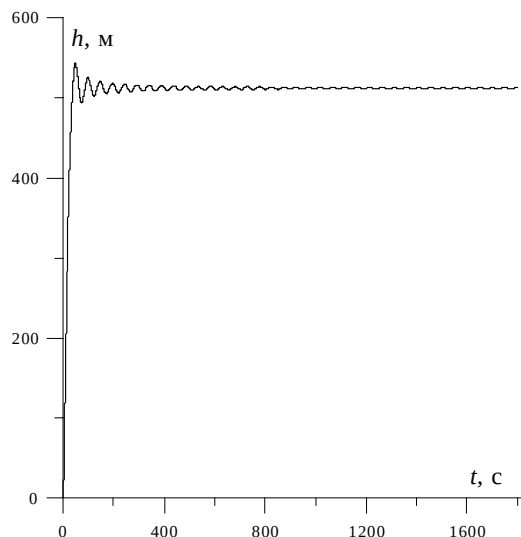


Рис.4. Графік висоти під'йому кулі.

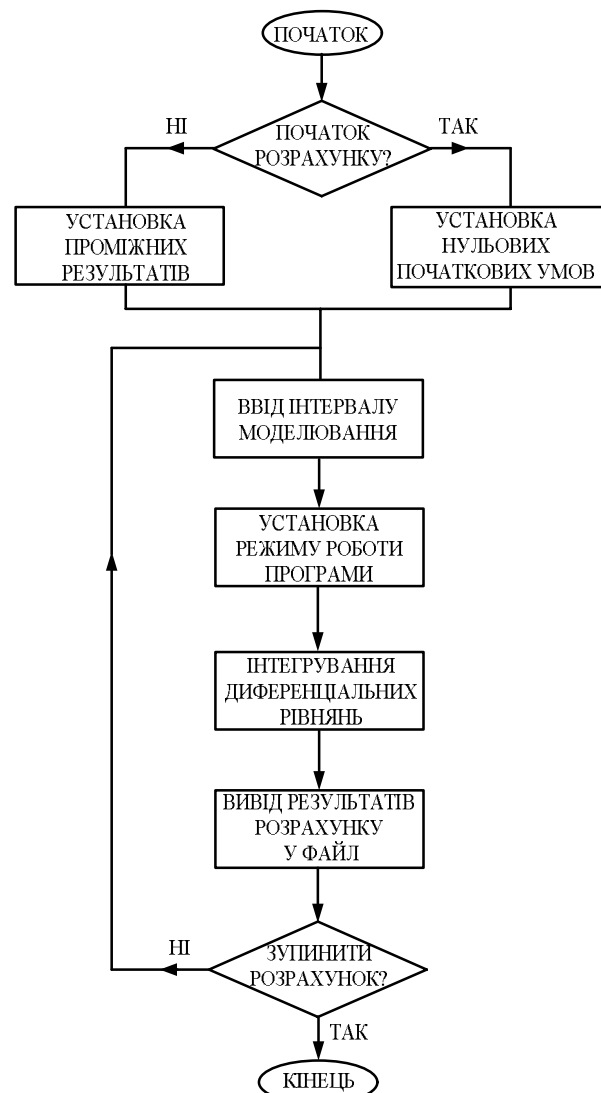


Рис.5. Блок-схема програми.

Усталене значення висоти кулі становить $h = 511$ м при $V = 0$ м/с. Слід зазначити, що незначне коливання висоти та швидкості в кінці інтервалу інтегрування зумовлене прийнятими входними допущеннями, похибкою методу, що в кінцевому результаті відповідають за похибку розрахунків, яка лежить в допустимих межах.

Система диференціальних рівнянь (9) становить модель піднімання повітряної кулі. Її переваги: диференціальні рівняння є безпараметричними і записаними в нормальній формі Коші.

Висновки. Комп'ютерна програма, виконана на основі розробленої математичної моделі за об'ємом пам'яті і обчислень проста, вигідно відрізняється можливістю здійснювати розрахунки з наперед заданою точністю. Розроблена математична модель дає можливість легко змінювати параметри досліджуваного об'єкту і тим самим аналізувати різні перехідні процеси. Так, при певних параметрах кулі вона може і не піднятися у повітря. Використання методів математичного моделювання та комп'ютерної симуляції дасть змогу відмовитись від натурних експериментів, які в багатьох випадках є складними і коштовними.

1. Бартеньев О. В. Современный Фортран / О. В. Бартеньев. – М. : Диалог-МИФИ, 1998. – 397 с.
2. Рыжиков Ю.И. Программирование на Фортране POWERSTATION для инженеров / Ю.И. Рыжиков. Практическое руководство. Санкт Петербург: КОРОНА, 2000, 161 с.
3. Хвищун І. О. Програмування і математичне моделювання: Підручник / І. О. Хвищун. – К. : Видавничий Дім "Ін Юре", 2007. – 544 с.
4. Чабан В. Чисельні методи / В. Чабан. Львів: В-во Нац. у-ту „Львівська політехніка”, 2001. – 186 с.

УДК 51:33

Лісковець С.М., Гуда О.В.

Луцький національний технічний університет

ПРОФЕСІЙНЕ СПРЯМУВАННЯ КУРСУ «ВИЩА МАТЕМАТИКА» ДЛЯ СТУДЕНТІВ ЕКОНОМІЧНОГО ПРОФІЛЮ

Лісковець С.М., Гуда О.В. Професійне спрямування курсу «Вища математика» для студентів економічного профілю. У статті розглядається застосування апарату лінійної та векторної алгебри, аналітичної геометрії, диференціального та інтегрального числення, функцій багатьох змінних, диференціальних рівнянь в економіці. В роботі зосереджена увага на важливості математичної освіти для сучасних спеціалістів економічного профілю.

Ключові слова: економічний профіль, лінійна алгебра, геометрія, похідна, інтегральне числення, диференціальні рівняння, математична освіта, економічні дослідження.

Лісковец С.М., Гуда О.В. Профессиональная направленность курса «Высшая математика» для студентов экономического профиля. В статье рассматривается применение аппарата линейной и векторной алгебры, аналитической геометрии, дифференциального и интегрального исчисления, функции многих переменных, дифференциальных уравнений в экономике. В работе сосредоточено внимание на важности математического образования для современных специалистов экономического профиля.

Ключевые слова: экономический профиль, линейная алгебра, геометрия, производная, интегральное исчисление, дифференциальные уравнения, математическое образование, экономические исследования.

Liskovec S., Guda O. Professional direction of "Higher Mathematics" course for students of economic profile. The article deals with the application of the apparatus of linear and vector algebra, analytic geometry, differential and integral calculus, the function of many variables, differential equations in the economy. The work focuses on the importance of mathematical education for modern specialists in the economic field.

Key words: economic profile, linear algebra, geometry, derivative, integral calculus, mathematical education, economic research.

Постановка наукової проблеми. Переорієнтація студентів з процесу навчання на практичний результат, на формування професійних складових, які необхідні спеціалісту в різних сферах його діяльності – одна із основних тенденцій у вищій школі. При цьому погляди сучасних студентів на навчальний процес все більше направлені на безпосереднє практичне застосування отриманих знань та навиків. Саме вирішення актуальних освітніх задач, поставлених часом, та прагнення студентів до отримання фахового багажу знань повинні бути пріоритетними при вивченні тих чи інших програмних дисциплін навчального процесу.

Виклад основного матеріалу. В контексті професійної економічної освіти курс «Вища математика» для економічних спеціальностей має важливе та багатогранне значення. Математична база є основою фахових сучасних дисциплін таких як, «Економіко-математичне моделювання», «Теоретична та прикладна статистика», «Економічна статистика» та багатьох спеціальних економічних курсів, зокрема – «Багатовимірний аналіз даних у контролі й аудиті», «Економіко-математичне моделювання фінансового стану підприємства», «Аналіз виробничо-господарської діяльності», «Комп'ютерна техніка та економічне програмування», тощо.

Базові математичні знання мають важливе значення для професійної підготовки фахівців економічного профілю, адже саме застосування математичних методів та моделей лежить в основі економічного аналізу. Вміння аналізувати, узагальнювати, робити висновки та генерувати ідеї, так необхідні економічним спеціалістам, – одне із основних завдань практичного застосування математичного апарату. Сучасний ринок праці потребує фахівців, які не тільки володіють теоретичними знаннями, але й вміють самостійно мислити, оцінювати обстановку та приймати оптимальні рішення в різних ситуаціях.

Серед шляхів підвищення мотивації до вивчення вищої математики студентами-економістами є широке застосування економічних задач в ході опрацювання теоретичного курсу математичних розділів. Для наглядного бачення студентами необхідності вивчення теорії потрібно в навчальний процес максимально включати завдання економічного змісту, які не тільки формують професійну зацікавленість, але й творчу установку на майбутню професію. Крім того, математична підготовка суттєво допомагає оцінювати та прогнозувати процеси, що відбуваються в сучасній економіці. Студенти-економісти повинні вміти проводити математичні обчислення та розрахунки, обробляти отримані результати та аналізувати їх на достатньо високому професійному рівні та, що дуже важливо, застосовувати набуті знання на практиці.

Для отримання відповідної математичної підготовки студенти економічних спеціальностей повинні оволодіти основними поняттями та положеннями багатьох розділів вищої математики, серед яких-лінійна та векторна алгебри, аналітична геометрія, математичний аналіз (диференціальне та інтегральне числення функції однієї змінної), функції багатьох змінних, диференціальні рівняння, тощо. Значимість даних розділів математики для професійної освіти підкріплюється різноманітними задачами економічного змісту, що мають широке практичне застосування та сприяють реалізації міжпредметних зв'язків у процесі вивчення вищої математики.

Однією із проблем, які виникають в процесі навчання студентів економічних факультетів, є невміння зв'язати математичну теорію з економічними процесами, оцінити її значення в ході вирішення фахових завдань. В багатьох випадках студенти не можуть уявити цілісну картину економіко-математичної ситуації, що приводить до відсутності інтересу при вивченні математики. Змінити відношення до вивчення дисципліни можна шляхом максимального використання задач прикладного характеру, які зможуть професійно зацікавити студентів та переконати їх у важливості математичної підготовки. При цьому слід зазначити, що кількість навчальних кредитів для вивчення курсу «Вища математика» на економічних факультетах необґрунтовано має тенденцію до зменшення, такий підхід особливо характерний для вищих навчальних закладів регіонального рівня, що не може сприяти отриманню якісної математичної підготовки.

Курс «Вища математика» для студентів економічного профілю доречно розпочинати із окремих фінансових задач, які нададуть можливість математичній підготовці з перших занять отримати професійне забарвлення. Наприклад, узагальнюючи тему «Арифметична та геометрична прогресії», яка необхідна для подальших економічних розрахунків, потрібно повторити обчислення простих та складних відсотків. Так, зокрема, зробити акцент на тому, що у випадку, якщо сума коштів P вкладена під R відсотків річних, то після першого року буде одержано

прибуток величиною $d = P \cdot \frac{R}{100}$. При цьому величина капіталу P , вкладеного під простий відсоток R , через n років обчислюється за формулою

$$a_n = P \left(1 + \frac{nR}{100} \right).$$

Такі актуальні формули – це наслідок найпростіших властивостей арифметичної прогресії. При цьому, формули, що мають відношення до складних відсотків, впливають із властивостей геометричної прогресії. Так, величина капіталу P , що зростає кожного року на R складних відсотків через n років прийме значення

$$P_n = P(1 + 0.01R)^n.$$

Основні формули, які часто застосовуються для економічних розрахунків, утворюють математику фінансів. Це, в першу чергу, підрахунок різних середніх величин та похибок, обчислення простих та складних відсотків прибутку; розрахунки накопичення, на які регулярно нараховується та зараховується фіксований дохід та робиться баланс вкладень і запланованих відсотків; розрахунки ренти; щорічної ренти; погашення боргу (процес повернення боргу регулярно, певними частинами, в певний термін і на протязі обумовленого часу із виплатою певного відсотка), тощо.

Вивчення «Лінійної алгебри» – першого розділу «Вищої математики», доречно поєднувати із розв'язуванням задач економічного напрямку, в яких широко використовуються основні поняття (матриці, визначники, обернені матриці, системи лінійних алгебраїчних рівнянь). Студенти-економісти повинні переконатися, що матриці широко застосовуються для планування виробництва та транспортних перевезень, в планових розрахунках, для розрахунків фонду заробітної плати. Матриці дозволяють розробити різні варіанти плану, провести відповідний аналіз, полегшують дослідження залежності між різними показниками. Так, модель міжгалузевого планування потреб та пропозицій базується на аналізі матриці потреб та пропозицій між різними галузями промисловості (при обчисленні майбутніх пропозицій використовуються елементи оберненої матриці). Серед основних економічних задач, які зводяться до розв'язування систем лінійних алгебраїчних рівнянь різними методами, відносяться такі завдання:

- задача знаходження коефіцієнтів повних та непрямих витрат, економічного плану та програми виробництва (застосовується обернена матриця, матричний спосіб розв'язування систем лінійних рівнянь),

- задача знаходження витрат сировини, палива та трудових ресурсів, при цьому розраховуються сумарні витрати для виконання програми виробництва та внутрішні виробничі витрати (використовується механізм множення матриць), тощо.

- задача прогнозування об'єму випуску продукції (результат розв'язку неоднорідної системи лінійних алгебраїчних рівнянь).

Апарат лінійної алгебри застосовується в спеціальних економічних курсах для складання моделі взаємної закупівлі товарів, для опису лінійної матричної моделі міжнародної торгівлі, для аналізу міжгалузевої економіки.

Вектори, як основний об'єкт вивчення векторної алгебри, знайшли різнопланове застосування в задачах мікроекономіки. Так, широко використовується вектор-рядок вартості $V = (v_1, v_2, v_3, v_4)$, елементами якого є вартості різної сировини, палива, робочої людиною –

години та вектор-стовпець $D = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}$ потреб відповідних галузей до продукції окремих ділянок

виробництва та багато інших. В даному розділі доцільно зробити акцент на продуктивну функцію, яка використовується при аналізі закономірностей виробництва і яка, по суті, є співвідношенням між використаними у виробництві ресурсами та випущеною продукцією. Із продуктивною функцією тісно пов'язана функція випуску та функція виробничих витрат. Ці функції у певних випадках використовують правила дій над векторами.

Найпростіші лінійні статистичні економічні системи описуються за допомогою векторів. Наприклад, для дослідження динамічних моделей різних процесів стан окремої економічної системи, що вивчається, в момент часу t описується за допомогою вектора X із n вимірного простору, а керування процесами в той самий момент часу описується за допомогою вектора $U(t)$ із k вимірного простору. Таким чином, в динамічних моделях використовують вектори n та k вимірних просторів, координати яких залежать від часу t . Можна зауважити, що курсі «Математичне програмування» та в окремих економічних дослідженнях використовується поняття опуклої лінійної комбінації векторів.

Цікавими та професійно важливими є задачі економічного змісту, в яких застосовуються основні поняття, методи та формули аналітичної геометрії. Найбільш широко використовуються рівняння прямої та кривих ліній на площині, рівняння площини, прямої, поверхонь в просторі та їх графіки. Основним методом аналітичної геометрії є метод координат, крім того, для розв'язування задач геометрії застосовуються методи лінійної та векторної алгебри. Серед задач економічного змісту, в яких використовуються графіки прямих на площині та їх рівняння з кутовим коефіцієнтом, можна розглядати такі завдання:

- визначення рентабельності транспортного постачання;
- визначення найкращого плану виробництва, якщо задаються певні витрати;
- визначення витрат палива (знаходиться аналітична залежність між витратами палива та швидкістю);
- дослідження впливу розширення парку сільськогосподарських машин на зростання врожайності зернових культур;
- визначення рівноваги доходу та збитків, та багато інших.

Варто зауважити, що задачі економічного змісту, наведені вище, не охоплюють весь об'єм практичного застосування теоретичних знань, розглянутих розділів вищої математики. Обсяг задач професійного спрямування можна значно збільшити. При цьому використання завдань з прикладним фаховим змістом, крім того, що з перших занять надає математиці професійного забарвлення, ще сприяє розширенню кругозору студентів-економістів, мотивує до накопичення широкого спектру математичних знань, вчить думати логічно та оригінально, шукати нові творчі підходи до розв'язання поставлених завдань. З іншої сторони, такий підхід допомагає зняти проблему формального вивчення математики, показує студентам тісний зв'язок математичних

понять і теорії з майбутньою професією, веде до якісної математичної підготовки спеціалістів економічного профілю.

Математичний аналіз займає не тільки центральне місце в базовій математичній підготовці, даний розділ – це важливим інструментом для дослідження різних процесів в економіці. Важливою задачею майбутніх спеціалістів є вивчення економічних величин, записаних у вигляді функцій. Враховуючи цей факт, під час вивчення основних властивостей математичних функцій доречно акцентувати увагу на ознайомленні та початковому аналізі функцій, що використовуються в економічних дослідженнях. В першу чергу, до таких функцій відносяться:

- функція попиту, споживання та пропозицій (залежність обсягу попиту, споживання, пропозицій стосовно окремих послуг або товарів від ціни, доходу, інших факторів),
- функція корисності (залежність результату, ефекту певної дії від інтенсивності цієї дії),
- виробнича функція (залежність результату виробничої діяльності від фактору, який сприяє його появі),
- функція випуску (залежність обсягу виробництва від наявності або споживання ресурсів),
- функція витрат (залежність витрат виробництва від обсягу продукції), тощо.

Дослідження багатьох економічних процесів зводиться до аналізу цих та інших функцій, який базується на вивченні їх граничних та диференціальних властивостей. Відповідно, початкове ознайомлення з такими об'єктами економічних досліджень буде досить актуальним. В процесі вирішення окремих професійних задач створюються математичні моделі, будуються відповідні функції, які аналізуються за допомогою апарату диференціального та інтегрального числення. Наприклад, до таких задач відносяться завдання про отримання максимального прибутку при мінімальних затратах; про збільшення або зменшення виручки фірми за умови підвищення ціни на її продукцію; про з'ясування напрямку зміни доходу держави за умови збільшення або зменшення тих чи інших податків.

Вивчаючи базову основу диференціювання функцій однієї змінної та зміст похідної, для студентів економічного профілю доречно розглядати, наприклад, задачу про маргінальну вартість, дохід, прибуток. Маргінальними витратами називають гранично можливі витрати в умовах хоча б постійного відтворення виробництва. Аналогічно визначаються маргінальні доходи та прибуток.

Якщо $V(x)$, $D(x)$, $P(x)$ – функції відповідно витрат, доходу, прибутку виробництва x одиниць продукції, то $V'(x)$, $D'(x)$, $P'(x)$ – маргінальні вартість, дохід, прибуток (похідна функції витрат – маргінальна вартість). Похідні функції $V'(x)$, $D'(x)$, $P'(x)$ представляють економічний зміст похідної функції однієї змінної. Так як, дослідження швидкості зміни одного параметра веде до поняття похідної, то можна розглядати ряд інших аналогічних прикладів економічного характеру. Наприклад, якщо на момент часу t_0 виробник виготовив $f(t_0)$ одиниць

продукції, то відношення $\frac{\Delta f(t_0)}{\Delta t}$, де Δf – приріст випущеної продукції за час Δt , називають середньою продуктивністю праці виробника за час Δt , а границю

$$\lim_{\Delta t \rightarrow 0} \frac{\Delta f(t_0)}{\Delta t} = f'(t_0) = p(t_0)$$

продуктивністю праці виробника в момент часу t_0 (економічний зміст похідної). Після акценту на економічний зміст похідної студентам фахового профілю не буде підстав сприймати диференціальне числення як формальну, відірвану від економіки, математичну тему. Крім того, в економіці широко використовуються середні величини: середній дохід, середній прибуток, середня продуктивність і так далі. Однак, частіше потрібно знати, як змінюється результат, якщо будуть збільшені витрати, або, навпаки, на скільки зменшиться результат, якщо витрати зменшаться. До студентів доречно довести, що в таких задачах потрібно визначити границю відношення приросту результату до витрат, тобто знайти похідну.

Для дослідження економічних процесів, які залежать від різних факторів, потрібно аналізувати функції багатьох змінних. Наприклад, попит на будь-який товар залежить від вартості його одиниці, якості, доходів споживачів, ціни альтернативного товару, пакування та інших чинників. Практичне застосування функцій кількох змінних стає очевидним. Вивчення властивостей таких функцій базується на дослідженні їх ліній та поверхонь рівня, на граничних та диференціальних властивостях. Питання про те, як змінюються значення відповідних виробничих

функцій при зміні одного із факторів веде до поняття частинних похідних. Важливим є поняття еластичності попиту, яка залежить від кількості виробів, які виготовлені та продані за певний інтервал часу, та від вартості одного виробу; при цьому еластичність виражається через частинні похідні. З економічної точки важливим є встановлення зв'язку між доходами підприємства та еластичністю попиту. Задача про кількісне значення об'єму випуску деякого товару, за який фірма отримає найбільший прибуток, зводиться до знаходження екстремуму функції кількох змінних. Аналогічною є задача про найменші витрати. Ці поняття та задачі економічного змісту професійно спрямовують студентів на вивчення теоретичної бази функцій багатьох змінних та є переконливими аргументами практичного використання математичних знань.

Вивчення інтегрального числення студентами економічного профілю доцільно також наповнювати прикладними задачами. Інтегрування в економіці має широке застосування, зокрема, для знаходження функцій витрат, прибутку, споживання, якщо відомі відповідно функції граничних витрат, граничного прибутку, граничного споживання, тощо.

Під час опрацювання матеріалу про обчислення означеного інтеграла та використання формули Ньютона-Лейбніца доречно розглядати, наприклад, задачу про витрати, дохід та прибуток. Нехай $V(x)$ – функція загальних витрат на виробництво x одиниць продукції, $V'(x)$ – функція маргінальних витрат. Тоді визначений інтеграл

$$\int_a^b V'(x)dx = V(x) \Big|_a^b = V(a) - V(b)$$

дорівнює зміні загальних витрат при зростанні кількості виробленої продукції від a до b одиниць. Звідси випливає такий висновок: зміна виробничих витрат при зростанні виробленої продукції від a до b одиниць дорівнює площі криволінійної трапеції, обмеженої графіком функції маргінальних витрат $y = V'(x)$, відрізком $[a, b]$ та прямими $x = a$, $x = b$.

Серед економічних задач, в яких використовується означений інтеграл:

- задача про коефіцієнт нерівномірного розподілу прибуткового податку (функція, яка описує дійсний розподіл прибуткового податку, називається кривою Лоренца);
- задача максимізації прибутку за певний проміжок часом;
- задача стратегії розвитку, зростання капіталу (дається відповідь на питання, яку кількісну стратегію потрібно обрати компанії, якщо номінальна облікова щорічна ставка відома).

При вивченні диференціального та інтегрального числення на економічних факультетах особливу увагу потрібно приділити таким поняттям, як границя, похідна, інтеграл, які широко використовуються в економічних дослідженнях.

Розділ «Диференціальні рівняння» – важливий апарат для дослідження різних процесів в економіці. Диференціальні рівняння застосовуються для знаходження закону зростання чисельності населення, функції попиту на товар (відомою є еластичність попиту) та в багатьох інших ситуаціях. Студентам-економістам потрібно показати необхідність вивчення даного розділу математики для вміння аналізувати економічні процеси. Наприклад, знайомство з найпростішими диференціальними рівняннями та задачею Коші можна почати задачею про зростання інвестицій. Встановлено, що швидкість зростання інвестованого капіталу у будь-який момент часу t пропорційна величині капіталу із коефіцієнтом пропорційності рівним узгодженому відсотку R неперервного зростання капіталу. Потрібно знайти закон зростання інвестованого капіталу, враховуючи величину початкової ($t = 0$) інвестиції K_0 . Будуємо математичну модель: $K(t)$ –

величина інвестованого капіталу в момент часу t (шукана функція), тоді $\frac{dK(t)}{dt}$ – швидкість зміни

величини інвестицій, $r = \frac{R}{100}$ – коефіцієнт пропорційності. За умови задачі маємо

$$\begin{cases} \frac{dK(t)}{dt} = rK(t), \\ K(0) = K_0 \end{cases}$$

Розв'язок рівняння – $K(t) = e^{rt+c} = e^c \cdot e^{rt}$. Згідно початкової умови, при $t = 0$ $K_0 = e^c$.

Отже, розв'язком задачі Коші є функція – $K(t) = K_0 e^{rt}$, а це означає, що інвестиції з часом зростають за експотенціальним законом.

Диференціальні рівняння широко використовуються при моделюванні проблем інфляції, економічного зростання, взаємозв'язків грошового та реального ринків та при вивченні різних закономірностей економічного розвитку суспільства.

Висновки: Таким чином, математика в системі підготовки сучасного економіста не може займати другорядне місце, так як є важливим апаратом для розв'язання прикладних задач та економічних досліджень. Для того, щоб процес вивчення математики позитивно впливав на рівень сформованих знань та навиків, необхідних студенту економічного профілю для фахових обов'язків, потрібно:

- спрямувати курс вищої математики на розв'язання професійно важливих задач;
- підвищити мотивацію студентів під час вивчення математики за рахунок включення в процес навчання задач економічного змісту;
- поглиблено вивчати лише той матеріал, який найбільш важливий для професійної діяльності;
- посилити зв'язок математики з майбутньою професією;
- поєднати математичні поняття із знаннями, необхідними для вирішення економічних завдань;
- отримані математичні знання, повинні доповнювати спектр професійної підготовки студентів економічного профілю.

1. Вища математика для економістів. Конспект лекцій /Уклад. Ю.П. Буценко, О.О. Диховичний, О.А. Тимошенко. – К: НТУУ «КПІ», 2014 – 256 с.
2. Барковський В.В. , Барковська Н.В. Вища математика для економістів. – Київ: ЦУЛ, 2010 – 400 с.
3. Васильченко І.П. Вища математика для економістів. – Київ: Знання, 2007 – 454с.
4. Долгих В.М. Вища математика для економістів. –ДВНЗ «Українська академія банківської справи Національного банку України», 2008– 110 с.

УДК 539.375

Matviiv Yu.Ya., S.D, Andrushchak I.E., S.D, Ganulich B.K., Ph.D, Kradinova T.A., Ph.D.
Lutsk National Technical University

PRACTICAL APPLICATION OF THE CRITERION FOR DETERMINING LONG-TERM STRENGTH AND RESIDUAL DURABILITY OF STRUCTURAL ELEMENTS AT LOW TEMPERATURE CREEP

Matviiv Y.Ya., Andrushchak I.E., Ganulich B.K., Kradinova T.A. Practical application of the criterion for determining the long-term strength and residual durability of structural elements for low-temperature creep. In the article, in practice, the previously formulated criterion for determining the long-term strength and residual longevity of structural elements with cracks in the case of a long static load is considered and its application to the examples of problems for beams of an open profile during their prolonged stretching and bending.

Key words: long-term strength, residual durability, open profile beam, low-temperature creep, period up to critical crack growth, stress intensity factor.

Матвій Ю.Я., Андрущак І.Є., Ганулич Б.К., Крадінова Т.А. Практичне застосування критерію для визначення довготривалої міцності та залишкової довговічності елементів конструкцій за низькотемпературної повзучості. У статті на практиці розглянуто сформульований раніше критерій для визначення довготривалої міцності та залишкової довговічності елементів конструкцій з тріщинами за довготривалого статичного навантаження та застосування його на прикладах задач для балок відкритого профілю за їх довготривалого розтягу та згину.

Ключові слова: довготривала міцність, залишкова довговічність, балка відкритого профілю, низькотемпературна повзучість, період докритичного росту тріщини, коефіцієнт інтенсивності напружень.

Матвиив Ю.Я., Андрущак И.Е., Ганулич Б.К., Крадинова Т.А. Практическое применение критерия для определения длительной прочности и остаточной долговечности элементов конструкций при низкотемпературной ползучести. В статье на практике рассмотрены сформулированный ранее критерий для определения длительной прочности и остаточной долговечности элементов конструкций с трещинами при длительной статической нагрузке и применение его на примерах задач для балок открытого профиля при их длительном растяжении и изгибе.

Ключевые слова: длительная прочность, остаточная долговечность, балка открытого профиля, низкотемпературная ползучесть, период до критического роста трещины, коэффициент интенсивности напряжений.

For the technical diagnostics of metal structures, as well as reinforced concrete engineering structures, where elements are used for long-term static loads, the establishment of periods between preventive examinations and defectoscopic control, and hence the determination of the need for their repair or replacement, and thus their reliable operation, are necessary reliable methods for predicting long-term durability and durability, especially in the presence of cracks. Exploring this problem is purely empirical, through experimental research, is technically rather difficult and not always possible in principle. Thus there is a need to create a reliable propagation theory and describe the processes of origin and distribution of low-temperature creep cracks, as well as methods for calculating long-term durability and residual resource of said structural elements.

The issue of long-term durability of thin-walled structural elements is currently devoted to many works, especially with experimental studies [1–4]. However, most of them consider defect-free elements of structures. In the works [5–6], based on the first law of thermodynamics for the delayed destruction of thin-walled structural elements for long-term cyclic and static loads, an energetic approach for estimating the period of subcritical growth of cracks for such loads is formulated. We apply this approach to the calculation of long-term strength of thin-walled elements of structures with cracks of low temperature creep.

Formulation of the problem. Consider a thin-walled element of a construction made of quasi-crimped material. Linear parameters b_l characterize the configuration of the element, and the power parameter p is the external load parameter applied at a reduced temperature when the plastic volumes of the loaded body are low temperature creep [2]. We believe that the methods of defectoscopy have established the absence in the element of the design of large (in comparison with its size) defects that do not exceed a certain value $2l_0$. The task is to determine the smallest value of the external load, during which the given time $t = t_*$ there will be no destruction of this element.

Similarly [7, 8], we assume that in the vicinity of the most intense point of this element there is a crack with a characteristic size $2l_0$. Such a defect can be a rectilinear fracture with the greatest length

$2l_0$. The problem can be solved by the proposed method and in the case of other defects (cavity, inclusion, etc.), but only defects of the type of cracks that are most dangerous are considered here.

By methods of the theory of elasticity we determine the main stresses σ_1, σ_2 at the point O of this element, considering it initially as defectless. We'll get

$$\sigma_1 = f_1(p, b_i), \quad \sigma_2 = f_2(p, b_i), \quad \eta_0 = \frac{f_2(p, b_i)}{f_1(p, b_i)}; \quad (1)$$

where $f_j(p, b_i)$ ($j=1, 2$) are well-defined functions.

Given the continuity of the stress tensor, and also the fact that the value l_0 is small, we will assume (while increasing only the margin of safety) that is in the vicinity of the point G with the smallest diameter D ($D \gg 2l_0$) around the point O here is a homogeneous stressed state with major stresses σ_1, σ_2 . Assume now that in the vicinity of the point O of this element there is a rectilinear length fraction $2l_0$ the most dangerous orientation relative to the direction of the main stresses σ_1, σ_2 . Since $D \gg 2l_0$, then, the presence of a crack in the length $2l_0$ of the body will not affect the stressed state on the line in the vicinity of the point G , that is, there is realized a two-tensile stress tension σ_1, σ_2 . Consequently, the stressed state in the vicinity of such a crack can be approximated (it can be shown that the inaccuracy obtained in this case will go to the margin of safety).

Plate with arbitrarily oriented crack. Consider an infinite perfectly elastic-plastic plate with a straight-line macrocrack of initial length $2l_0$, which extends over the infinity of uniformly distributed forces of intensity in mutually perpendicular directions at an angle α to the line of the crack (Fig. 1), as well as for an unlimited plate with such a defect with a two-wise stretching effort σ_1, σ_2 . We believe that in the plastic zones near the top of the crack there is a phenomenon of low temperature creep. Determine the parameters of the external load $\sigma_1 = \sigma_{1*}(t_*)$, $\sigma_2 = \sigma_{2*}(t_*)$, in which the residual durability of the plate will not exceed the specified value $t = t_*$.

Since this problem is reversed to the problem of determining the period of subcritical growth of a low-temperature creep crack $t_* = t_*(\sigma_{1*}, \sigma_{2*})$, we first consider the direct problem.

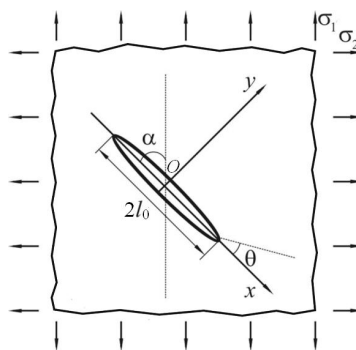


Fig. 1. – Scheme load of the plate with an arbitrarily oriented crack

As shown in [7, 8], the maximum intensity of stress at the vertices of the crack is achieved at $\alpha \approx \pi/2$ at $\eta_0 < 1$ ($\eta_0 = \sigma_1^{-1}\sigma_2$) and, analogically, $\alpha = 0$, $\eta_0 > 1$. From the results of papers [7, 8] it follows that this corresponds to the maximum value of the rate of propagation of the crack $dl/dt = V_{\max}$. So, for orientation $\alpha = \pi/2$ for $\eta_0 < 1$ and $\alpha = 0$ for $\eta_0 > 1$, the fracture and the least durability of the plate will be most dangerous. Find for these cases $t = t_*$. Based on the results of [7-10] we obtain an equation for determining the period of subcritical growth of a low temperature creep crack:

$$\frac{dl}{dt} = \frac{A_{2t} K_{CC}^{-2m} (K_I^{2m} - K_{thc}^{2m})}{1 - K_{CC}^{-2} K_I^2},$$

(2)

under initial and final conditions

$$t = 0, l(0) = l_0; \quad t = t_*, l(t_*) = l_*; \quad K_I(F, l_*) = K_{CC}. \quad (3)$$

Here A_{2t}, m are the characteristics of low temperature creep; $K_{C\tilde{N}}$ is critical value of K_I ; $K_I(F, l) = F\sqrt{\pi l}$.

Integrating equation (2) under initial and final conditions (3), we obtain

$$t_* = \frac{K_{CC}^{2(m-1)}}{A_{2t}} \int_{l_0}^{l_*} \frac{K_{CC}^2 - \pi l F_*^2}{(\pi l F_*^2)^m - K_{thc}^{2m}} dl, \quad F_* = \begin{cases} \sigma_{*1}, & \alpha = \pi/2, \quad \eta_0 < 1; \\ \sigma_{*2}, & \alpha = 0, \quad \eta_0 > 1. \end{cases} \quad (4)$$

Assuming that $l_* \gg l_0$, $K_{CC} \gg F_* \sqrt{\pi l_0}$, formula (4) can be approximated to represent this way

$$t_* \approx \frac{l_0 z}{A_{2t}} \left[\frac{1}{m-1} - \frac{z K_{thc}^{2m}}{K_{CC}^{2m} (2m-1)} \right], \quad z = \left(\frac{K_{C\tilde{N}}^2}{\pi l_0 F_*^2} \right)^m \quad (5)$$

From here

$$F_* \approx \frac{K_{CC}}{\sqrt{\pi l_0}} \left\{ t_* l_0^{-1} A_{2t} (m-1) \left[1 + \frac{t_* A_{2t} K_{thc}^{2m} (m-1)^2}{l_0 K_{CC}^{2m} (2m-1)} \right] \right\}^{-\frac{1}{2m}}. \quad (6)$$

On the basis of the relations (4) – (6) for the determination of critical values $\sigma_{1*} = \sigma_{1*}(t_*)$, $\sigma_{2*} = \sigma_{2*}(t_*)$, we write the following formulas:

$$\sigma_{1*} = F_*, \quad \eta_0 < 1; \quad \sigma_{2*} = F_*, \quad \eta_0 > 1, \quad (7)$$

where $\eta_0 = \sigma_{1*} \sigma_{2*}^{-1}$.

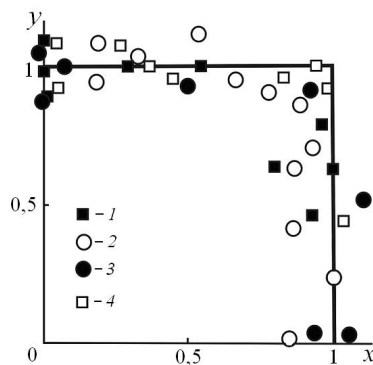


Fig.2. – Diagram of boundary loads for the plate: solid line - ratio (7.8); experiments: 1 – steel [11]; 2–4 – pig-iron samples in different states (2 – $\sigma_t = 345,3 \text{ MPa}$; 3 – $\sigma_t = 185,4 \text{ MPa}$; 4 – $\sigma_t = 228,6 \text{ MPa}$) [12]

We will make the following replacements in (7) $x = \sigma_{1*} q_*^{-1}$, $y = \sigma_{2*} q_*^{-1}$, де q_* – critical importance extending efforts q for a one-size-fits-all load. Then (7) can be written in this form

$$x = 1, \quad (\alpha = \pi/2, \quad \eta_0 < 1), \quad y = 1, \quad (\alpha = 0, \quad \eta_0 > 1). \quad (8)$$

Based on the dependences (8) in Fig.2, diagrams of boundary loads for a plate with a crack are constructed. Experimental results of low-temperature strength with two-wise stiffness of thin-walled elements without cracks are also presented here [11, 12], which are well in agreement with the results (8). Built on fig. 2 diagram of the boundary loads for plates with cracks and is the basis for calculating the low-temperature strength of thin-walled elements of structures with cracks.

The criterion of long-term residual strength. If the boundary-equilibrium state of the vicinity G is found, then the critical parameter of the external load $p = p_*$ is calculated from the condition

$$f_1(p_*, b_i) = F_*, \quad (9)$$

which will give the lower (most dangerous with the defect of the material) the value of the marginal load. We write the relation (9) in this form

$$\begin{aligned} \sigma_{1*} - F_* &= 0, & \eta_0 \leq 1; (\eta_0 = \sigma_{2*} \sigma_{1*}^{-1}) \\ \sigma_{2*} - F_* &= 0, & \eta_0 \geq 1 \end{aligned} \quad (10)$$

which is the equation of the boundary stress diagram in the Cartesian coordinate system $O\sigma_{1*}\sigma_{2*}$ (see also Fig.2). The diagram (10) limits the area of values of the main stresses σ_1, σ_2 , safe in relation to the strength of the structural element containing defects of this type during the specified lifetime $t = t_*$ of its operation. Taking into account this, as well as using the relation (10), we obtain the following condition of strength of quasi-violent bodies:

$$\sigma_{1*}(p_{i*}) - F_*(l_0, t_*, \eta_0) < 0, \quad (\eta_0 = \sigma_{1*}^{-1} \sigma_{2*}). \quad (11)$$

where the value F_* is determined in the (6).

In engineering practice, cases where initial defects in structural elements are small, but have different configurations (non-linear, surface non-transverse, etc.) are often encountered. To determine the long-term durability of structural elements in such cases, repeat the procedure described above, but for the defects of the given configuration. If the initial defects in the structural elements are not small and are proportional to the dimensions of the structural element, then to determine the residual durability of such elements during a predetermined period of its operation, we solve the direct problem (determination of the residual durability of the structural element with a crack) and the equation thus obtained for t_* (see, for example, (5)) we determine the critical value of the external load.

Determination of the acceptable size of the original defects. Along with the definition of the long-term strength of structural elements with cracks for engineering practice, it is important to establish the admissible dimensions of such defects when their resource and workload are specified. We solve this problem in the following way. We repeat all the considerations that are given at the beginning of this section up to the relation (5). Then from the relation (5), assuming that $K_{thc}^{2m} (\pi l_0 F_*^2)^{-m} \ll 1$, we determine the approximate permissible value of the size of the initial defect $l_{0*} = l_0(t_*, F_*)$, in which the given resource t_* is provided, i.e.

$$l_{0*} \approx \left(\frac{K_{CC}^2}{\pi F_*^2} \right)^{\frac{m}{m-1}} [A_{2t} t_* (m-1)]^{\frac{-1}{m-1}}. \quad (12)$$

If the defects a_i in the configuration parameters are not transverse and not straightforward, or large, then to determine their admissible sizes $a_{i*} = a_i(t_*, F_*)$, we solve the direct problem, that is, we construct a formula of type (5), from which we find a_{i*} .

Determination of long-term durability of open profile beams with cracks for their tensile strength. Consider the beams of the open profile of the channel (Fig. 6) and the corner (Fig. 10), which are weakened by cracks and are stretched by the long-term efforts of P . The problem is to find the largest allowable values of effort $P = P_*$ that will provide the residual resource l_0 at given initial sizes of cracks $t = t_*$.

The solution of this problem is carried out similarly to the above. For this we find the relation of type (5) in the case of each profile of the beams. The relations (7.28) and (7.37) for cases of a channel and a corner can be approximated (believe that, $l_* \gg l_0, K_{CC} \gg 1, 12 P_* \sqrt{\pi l_0}$) to represent this way

$$t_{i*} \approx \frac{1.25 l_0 K_{CN}^{2m}}{A_{2t} (1, 25 \pi l_0 P_i^2 F_i^{-2})^m} \left[\frac{1}{m-1} - \frac{K_{thc}^{2m}}{(1, 25 \pi l_0 P_i^2 F_i^{-2})^m (2m-1)} \right] (i = 1; 2.). \quad (13)$$

Here $i=1$ corresponds to the case of a channel, and $i=2$ – to the case of a corner. Solving approximately the equation (13), we obtain the following formula for determining the permissible value of effort $P = P_*$

$$P_* \approx F_i \frac{K_{CC}}{1,12\sqrt{\pi l_0}} \left\{ 0,8 t_* l_0^{-1} A_{2t}(m-1) \left[1 + \frac{t_* A_{2t} K_{thc}^{2m} (m-1)^2}{1,25 l_0 K_{CC}^{2m} (2m-1)} \right] \right\}^{-\frac{1}{2m}}. \quad (14)$$

Assuming that the beams are made of 10HCHF steel (the characteristics of low temperature creep are determined by the relations (7.24)) and the initial length of the cracks is equal to 0.005 m, the relation (14) can be written even more

$$p_{i*} = P_* F_i^{-1} \approx 465,2 \left\{ 2,9 t_* + 18,2 \cdot 10^{-5} t_*^2 \right\}^{-0.125} \quad (0 < t_* < 2 \cdot 10^4) \quad (15)$$

on fig.3 according to the formula (15) graphical dependence of the average value p_{i*} of admissible efforts P for the beams of the separation of the channel and the angle on the value of the residual resource t_* is constructed.

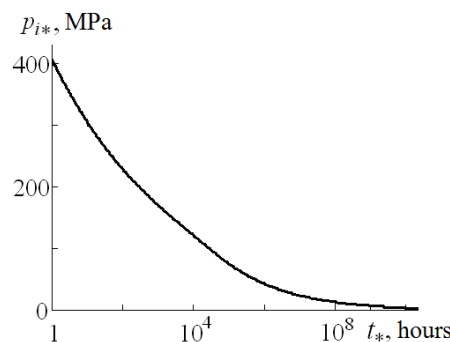


Fig.3. – Dependence of the parameter of the allowable load value p_{i*} from the given residual resource t_*

Estimation of allowable sizes of initial defects in open profile beams for their bending. We believe that the open-beam beams of the brand (Fig.13) and the two-javelin (Fig.15) are weakened by cracks of initial length $l = l_0$ and exposed to long-term (at the time $t = t_*$) bending moments M . The problem is to determine the greatest magnitude of the initial crack $l_0 = l_{0*}$, in which the beams will not be destroyed during the time $t = t_*$ given a given long-term load of moments M .

This task is to some extent reversed to determine the residual resource of such beams. The general approach to an approximate solution to such a problem is presented above. According to this approach, it is necessary to construct a closed solution of an approximate solution of the direct problem as an equation of type (5) and to find an unknown value $l_0 = l_{0*}$ from it. To do this, we will analyze the solutions of the direct tasks for the brand and the two-tauren, which are given in the second paragraph of this section. On the basis of this, and also considering that $l_* \gg l_0$, $K_{CC} \gg 1,12 M W_{xi}^{-1} \sqrt{\pi l_0}$, the solution of a direct problem can be approximated to be written as follows

$$t_{i*} \approx \frac{1,25 l_0 K_{CN}^{2m}}{A_{2t} (1,25 \pi l_0 M^2 W_{xi}^{-2})^m} \left[\frac{1}{m-1} - \frac{K_{thc}^{2m}}{(1,25 \pi l_0 M^2 W_{xi}^{-2})^m (2m-1)} \right] \quad (i = 1; 2). \quad (16)$$

Here W_{xi} is a moment of resistance of the cross-section when bending the beam relative to the axis xx if $i=1$ for the brand and if $i=2$ for the two-turret. Then from the relation (16), assuming that $K_{thc}^{2m} (1,25 \pi l_{0i} M W_{xi}^{-1})^{-m} \ll 1$, we determine the allowable value of the size of the initial defect $l_{0i} = l_{0i*}$, in which the given resource t_* is provided, that is

$$l_{0i*} = \left(\frac{K_{CC}^2}{\pi M^2 W_{xi}^{-2}} \right)^{\frac{m}{m-1}} [A_2 t_* (m-1)]^{\frac{-1}{m-1}}. \quad (17)$$

For the case where the beams are made of steel 10XCHД (characteristics of low temperature creep are determined by the relations (7.24)) and the load of beams is equal to $p = MW_{xi}^{-1} = 170$ MPa, the relation (7.17) can be recorded even more

$$l_{0i*} = 4,6 \cdot 10^{-2} \cdot t_*^{-1/3}; \quad (0 < t_* < 10^4). \quad (18)$$

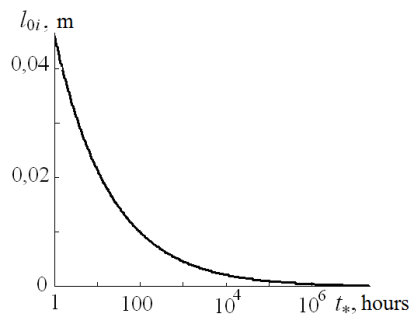


Fig. 4. - Graphical dependence of the allowable value of the size of the initial defect $l_{0i} = l_{0i*}$ on the value of the residual resource t_*

By the formula (18) in Fig.4 graphical dependence of the allowable value of the size of the initial defect $l_{0i} = l_{0i*}$ on the value of the residual resource t_* is constructed.

Conclusions and perspectives of further research. The criterion of long-term strength of elements of structures with cracks for their long-term static load and local low temperature creep is formulated. On the basis of this criterion, the long-term strength of the open-beam beams from the steel 10XCHД (channel, corner) was determined for their tensile strength and the presence of rectilinear cracks. It also determines the admissible sizes of cracks in the taurus and twin taurus due to their long bending and specified service life. The dependence of long-term durability on the planned resource of beams is analyzed.

1. Тайра С. Теория высокотемпературной прочности материалов / С. Тайра, Р. Отани. – М.: Металлургия, 1986. – 280 с.
2. Garofalo F. Fundamentals of creep and creep-rupture in metals/ F. Garofalo. – New York-London: Mac Millan Company, 1970. – 343p.
3. Андрейків О.Є. Математична модель для визначення періоду докритичного поширення тріщин високотемпературної повзучості в твердих тілах / О.Є. Андрейків, Н.Б. Сас // Доповіді НАН України – 2006. – №5. – С. 47–52.
4. Лепин Г.Ф. Ползучесть металлов и критерии жаропрочности / Г.Ф. Лепин. – М.: Металлургия, 1976. – 375 с.
5. Надаи А. Пластичность и разрушение твёрдых тел. Т. 2. / А. Надаи. — М.: Мир. – 1969. – 863 с.
6. Андрейків О.Є. Циклічна міцність тонкостінних елементів конструкцій з тріщинами /О.Є. Андрейків, Ю.В. Банахевич, М.Б. Кіт // Доповіді НАН України. – 2009. – № 7. – С. 56–62.
7. Андрейків О.Є. Математична модель для визначення періоду докритичного поширення тріщин високотемпературної повзучості в твердих тілах / О.Є. Андрейків, Н.Б. Сас // Доповіді НАН України – 2006. – №5. – С. 47–52.
8. Андрейків О.Є. Механіка руйнування металічних пластин при високотемпературній повзучості / О.Є. Андрейків, Н.Б. Сас // Фіз.-хім. механіка матеріалів – 2006. – №2. – С. 62–68.
9. Довговічність пластин з тріщинами за довготривалого статичного навантаження і локальної повзучості /О.Є. Андрейків, В.Р. Скальський, Ю.Я. Матвіїв, Т.А. Крадінова // Фіз.-хім. механіка матеріалів. – 2012. – №1. – С 39–46.
10. Андрейків О.Є. Визначення довговічності пластин з системами тріщин в умовах дії довготривалого статичного розтягу і низькотемпературного поля / О.Є. Андрейків, Ю.Я. Матвіїв, Т.А. Крадінова // Мат. методи і фіз.-мех. поля. – 2011. – №4. – С.143–150.
11. Писаренко Г.С. Сопротивление материалов деформированию и разрушению при сложном напряженном состоянии /Г.С. Писаренко, А.А. Лебедев. – К.: Наук. думка, 1969. – 301 с.
12. Cornet J. Theories of Fracture Under Combined Stresses / J. Cornet, R.C. Crassi // Trans. ASME, Ser. D. – 1961. – Vol. 83, No1. – P. 39–44.

УДК 681.32

Молнар О.О.¹, Бан Г.Й.¹, Гал Д.Л.¹, Герасимов В.В.²,

¹ Ужгородський національний університет

² Мукачівський державний університет

СИСТЕМА ТЕЛЕМЕТРІЇ З ЕЛЕМЕНТАМИ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ СПІВРОБІТНИКІВ СЛУЖБ ПОРЯТУНКУ НА ОСНОВІ ПЛІС

Молнар О.О., Бан Г.Й., Гал Д.Л., Герасимов В.В. Система телеметрії з елементами доповненої реальності для співробітників служб порятунку на основі ПЛІС. В даній роботі описана система телеметрії для співробітників служб порятунку, яка визначає фізіологічний стан людини (температуру тіла, ЕКГ; частоту серцевих скорочень, рівень кисню в крові, опір шкіри, частоту та глибину подиху), фізичні параметри оточуючого середовища (температуру повітря, атмосферний тиск, вологість, рівень радіації (α , β , γ), рівень ультрафіолетового випромінювання, рівень шкідливих та небезпечних газів), виводить отримані дані в поле зору пожежника за допомогою окулярів доповненої реальності та передає їх по бездротовому каналу Wi-Fi в центр спостереження. Система побудована на основі модуля DE10-Nano Kit компанії Terasic яка містить ПЛІС Intel/Altera Cyclone® V SE 5CSEBA6U23I7.

Ключові слова: Датчики, фізичні параметри оточуючого середовища, фізіологічний стан людини, система доповненої реальності, ПЛІС.

Молнар А.А., Бан Г.Й., Гал Д.Л., Герасимов В.В. Система телеметрии с элементами дополненной реальности для сотрудников служб спасения на основе ПЛИС. В данной работе описана система телеметрии для сотрудников служб спасения, которая определяет физиологическое состояние человека (температуру тела, ЭКГ, частоту сердечных сокращений, уровень кислорода в крови, сопротивление кожи, частоту и глубину дыхания), физические параметры окружающей среды (температуру воздуха, атмосферное давление, влажность, уровень радиации (α , β , γ), уровень ультрафиолетового излучения, уровень вредных и опасных газов), выводит полученные данные в поле зрения пожарного с помощью очков дополненной реальности и передает их по беспроводному каналу Wi-Fi в центр наблюдения. Система построена на основе модуля DE10-Nano Kit компании Terasic которая содержит ПЛИС Intel/Altera Cyclone® V SE 5CSEBA6U23I7.

Ключевые слова: Датчики, физические параметры окружающей среды, физиологическое состояние человека, система дополненной реальности, ПЛИС.

Molnar A.A., Ban H.Y., Gal D.L., Gerasimov V.V. Telemetry system with elements of augmented reality for the employees of emergency services based on FPGA. In this paper, the results of developing a personal telemetry system for firefighters are presented. It consists of two subsystems - measuring the parameters of human activity and determining the parameters of the environment. All received data, together with warnings of danger, are displayed in the field of view of the firefighter using AR glasses, and also through the Wi-Fi interface are transferred to the center of remote diagnostics. The system is based on the Terasic DE10-Nano Kit, which contains the FPGA Intel Cyclone® V SE 5CSEBA6U23I7.

Keywords: Sensors, physical parameters of the environment, physiological state of the person, system of complemented reality, FPGA.

Згідно з останніми аналітичними звітами протягом наступних 10 років більше 100 мільйонів людей будуть носити одяг з вбудованими засобами індивідуальної телеметрії. Розробка подібних пристроїв ведеться під егідою впровадження так званих «Інтернет речей» (IoT - Internet of Things) [1]. Якщо оснастити одяг безліччю датчиків (температури, тиску, вологості, пульсу, електрокардіограми, рівня кисню в крові, опору шкіри і т. д.) і підключити їх до керуючого мікроконтролера з бездротовим каналом зв'язку Wi-Fi або Bluetooth LE, отримаємо систему дистанційної діагностики. Подібне екіпування дуже корисне для стеження за станом здоров'я спортсменів, самотніх людей похилого віку, дітей з особливими потребами, співробітників служб порятунку або пожежних команд.

Система телеметрії

Розроблена система складається з трьох основних модулів (рис.1). Модуль 1 на основі hSensor, або MAXREFDES100# фірми Maxim Integrated [2], визначає фізіологічні параметри людини, і закріплюється за допомогою спеціального реміня на грудині співробітника. В еластичний ремінь інтегровані електроди системи ЕКГ та датчик деформації грудної клітини. Оброблені дані по каналу Bluetooth LE передаються на модуль 2, пристрій збору, аналізу та збереження інформації (на основі DE10-Nano Kit фірми Terasic [3] побудованого на основі FPGA Cyclone® V компанії Intel/Altera), і через інтегрований World Wide Web (Web) – сервер, по інтерфейсу Wi-Fi (з використанням модулю RFS компанії Terasic [4]), передаються в центр спостереження. Локальний захищений накопичувач Secure Digital (SD) виконує функцію персональної «чорної скриньки» на аварійні випадки втрати зв'язку та для довготривалого збереження. В шоломі пожежника або співробітника служби порятунку вбудований цифровий

пірометр MLX90614ESF-DCI-000-SP фірми Melexis [5], який дистанційно визначає температуру об'єкту, на який дивиться співробітник, а виміряне значення поступає в модуль 2. Всі отримані та попередньо оброблені дані передаються по інтерфейсу Bluetooth LE на окуляри доповненої реальності фірми Intel/Recon Jet Pro. Ці значення відображаються в полі зору оператора, накладаючись на зображення реального світу (технологія Доповненої Реальності – Augmented Reality).



Рисунок. 1. Блок-схема системи телеметрії з елементами доповненої реальності. 1 - модуль спостереження за фізіологічним станом, 2 – модуль аналізу оточення, 3 – окуляри доповненої реальності Intel/Recon Jet Pro

Сюди же виводяться попередження про небезпеки, які виявляються системою (радіацію, небезпечні гази, стан здоров'я і т. д.). В умовах обмеженої видимості на окуляри можуть також передаватись креслення приміщення, в якому знаходиться співробітник, з вказівкою місцезнаходження людини і/або членів команди. Дані про його положення отримуються від 9 координатного ІВП - Інерційного вимірювального пристрою (IMU - Inertial measurement unit), вбудованого в кожен модуль системи, а на відкритій місцевості від модуля системи глобального позиціонування (GPS- Global Positioning System), вбудованого в окуляри.

Всі елементи системи можуть бути використані як окремо, так і в будь-якій комбінації. Наприклад модуль 1 - збору інформації про фізіологічний стан, може бути корисним для стеження за самотніми людьми похилого віку, або людей з особливими потребами. При виникненні небезпеки стану здоров'я, інформація може бути передана дільничному лікарю, в соціальну службу або в службу швидкої допомоги. Модуль 2 може бути корисним для шахтарів, військових і т. д., та навіть як окремо розташована метеостанція. А окуляри доповненої реальності мають просто необмежену сферу застосувань, багато з яких ще навіть не придумані.

Модуль спостереження за фізіологічним станом

Орієнтуючись на потреби розробників мобільних електронних пристроїв для спостереження за показниками фізичної активності користувача, компанія Maxim Integrated Products розробила платформу hSensor (MAXREFDES100#) [2]. На мініатюрній платі hSensor знаходиться одно каналний аналоговий інтерфейс для перетворення біо-потенціалів (електрокардіографію (ЕКГ), електроміографію (ЕМГ) або електроенцефалографію (ЕЕГ)) в цифровий код - MAX30003, безконтактний датчик частоти серцевих скорочень (ЧСС) і рівня насичення крові киснем MAX30101, датчик температури MAX30205 (його точність відповідає специфікації клінічної термометрії ASTM E1112, 0.1°C в межах від 37°C до 39°C), мікроконтролер з пониженим енергоспоживанням MAX32620 на ядрі ARM Cortex-M4F і контролер живлення MAX14720.

MAX30003 забезпечує формування в цифровому форматі кривої електрокардіограми (ЕКГ) та визначення частоти серцебиття. Його вхідний аналоговий канал оснащений захистом від

електростатичного розряду, виконує фільтрацію від електромагнітних завад, виявляє наявність постійного струму, має високий вхідний імпеданс, низький рівень шуму, високий коефіцієнт послаблення синфазної складової, різні варіанти програмованих фільтрів низьких та високих частот, а також аналого-цифровий перетворювач високої роздільної здатності. М'яка послідовність включення живлення гарантує, що на вхідні електроди не потрапляють великі сигнали від перехідних процесів. MAX30003 здатний працювати з великими напругами підключеними до електродів, і має швидкий режим відновлення, для миттєвої регенерації після перевантаження, наприклад після дефібриляції або електрохірургічного втручання.



а.



б.

Рисунок. 2. Зовнішній вигляд модуля hSensor а. (реальні розміри майже вдвічі менші ніж на фото), можливий варіант програми представлення отриманої інформації б. на платформі Андроїд.

Платформа hSensor, за рахунок мікросхеми MAX30101 підтримує різні типи вимірювання фотоплетизмографії на відбивання, включаючи пульс-оксиметрію та ЧСС на трьох довжинах хвиль - 880 нм (інфрачервоний, ІЧ), 660 нм (червоний) та 537 нм (зелений).

Крім того, оснащення hSensor включає інерційні датчики (які можуть бути використані для визначення положення та руху людини в просторі), барометричний сенсор атмосферного тиску (його показання легко перетворюються в висоту над рівнем моря з точністю ± 20 см, що може служити датчиком падіння людини), флеш-пам'ять програм та даних і інтерфейс Bluetooth LE. До складу платформи входить плата налаштування, драйвери, графічний інтерфейс користувача і вбудоване ПО, доступне в початкових кодах.

Модуль аналізу оточення

В якості системи обробки та доповнення отриманих даних додатковою інформацією (рівень радіації, ультрафіолетового опромінення, наявності шкідливих газів і т. д.) був розроблений модуль 2 на основі DE0-Nano Kit фірми Terasic [3] побудованого на основі програмованої логічної матриці (FPGA) Cyclone® V компанії Intel/Altera доповненої блоком розширення RFS тої же компанії.

DE0-Nano - малогабаритна плата для розробників. Його відмінні особливості:

- Intel Cyclone® V SE 5CSEBA6U23I7, (110K логічних елементів LE);
- Встановлена пам'ять EPCS16 для збереження конфігурації;
- Вбудований USB-Blaster II для програмування; JTAG режим;
- HDMI TX, сумісний з DVI 1.0 і HDCP v1.4;
- Три 50 МГц джерела тактових сигналів від тактового генератора;
- Два 40-контактних роз'єми розширення;
- Один роз'єм розширення Arduino (Uno R3 сумісний);
- Один 10-контактний роз'єм розширення входу аналогових сигналів (спільно з аналоговим входом Arduino);

- АЦП, 4-контактний SPI інтерфейс з FPGA.

HPS (Hard Processor System) і підключені до неї компоненти:

- 800 МГц двох ядерний ARM Cortex-A9 процесор;
- 1 ГБ DDR3 SDRAM (32-бітна шина даних);
- 1 Гбіт Ethernet з роз'ємом RJ45;
- USB OTG порт, USB Micro-AB роз'єм;
- Слот для Micro SD карт;
- Акселерометр (I2C інтерфейс + переривання);
- Міст UART - USB, USB Mini-B роз'єм;
- LTC 2x7 роз'єм розширення (для плат розширення Linear Technology).

Даний модуль об'єднує вбудований двох ядерний процесор ARM® Cortex™ -A9 з програмованою логікою для максимальної гнучкості дизайну.

В свою чергу плата RFS призначена для бездротового зв'язку, моніторингу навколишнього середовища і розробки додатків IoT (Internet of Things) [4]. Він може використовуватися з платою FPGA серії DE для розширення його функціоналу. RFS використовує 2x20 GPIO для взаємодії з платою FPGA. Обладнаний модулями бездротового зв'язку Wi-Fi (відстань до 100 метрів) та Bluetooth (відстань до 10 метрів). Крім того він оснащений декількома додатковими датчиками: 9-осьовим IMU: акселерометром, гіроскопом, магнітометром, датчиком освітлення, датчиком вологості та температури.

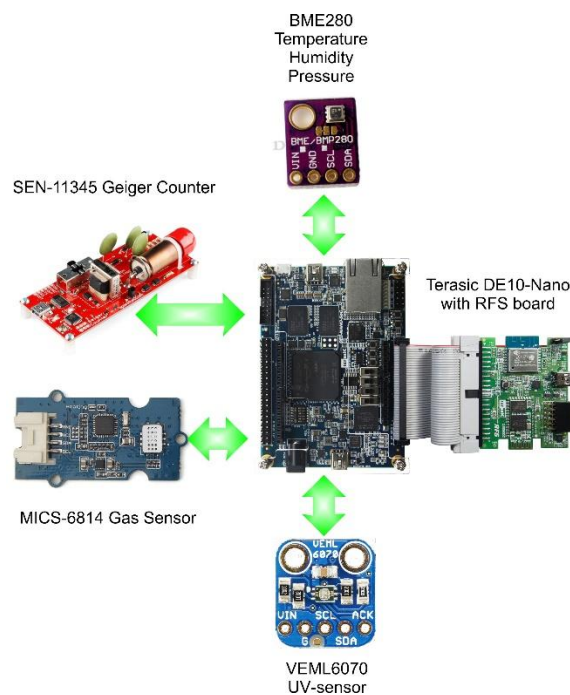


Рисунок. 3. Блок схема модуля моніторингу оточення.

Датчики

Під час вибору датчиків перевага була надана простим, легко доступним, дешевим і, що найважливіше, з низьким або наднизьким енергоспоживанням пристроям. В якості датчика температури, тиску та вологості, ми використали BME280 від Bosch Sensortec [6]. BME280 є інтегрованим датчиком навколишнього середовища, розробленим спеціально для мобільних пристроїв, де розмір та низьке енергоспоживання є ключовими обмеженнями дизайну. Він поєднує в собі високу лінійність, високоточні датчики тиску, вологості та температури, призначені для низького споживання струму ($3.6 \mu\text{A} @ 1 \text{ Гц}$), довготривалої стабільності та високої завадостійкості. Інтервал вимірювання тиску $300 \dots 1100 \text{ ГПа}$ з відносною точністю $\pm 0,12 \text{ ГПа}$ та роздільною здатністю $0,18 \text{ Па}$. Інтервал вимірювання температури $-40 \dots 85 \text{ }^\circ\text{C}$ з відносною точністю $\pm 1,0 \text{ }^\circ\text{C}$ та роздільною здатністю $0,01 \text{ }^\circ\text{C}$. Інтервал вимірювання вологості $0 \dots 100\%$ з відносною точністю $\pm 1\%$ та роздільною здатністю 0.008% .

Для визначення рівня ультрафіолетового випромінювання ми вибрали VEML6070 від Vishay [7]. Це сучасний датчик ультрафіолетового (УФ) випромінювання з інтерфейсом I²C. Застосована в VEML6070 технологія FiltronTM забезпечує найкращу спектральну чутливість для охоплення всього ультрафіолетового спектру. В режимі очікування споживання енергії менше 1 мкА.

Як сенсор α , β і γ часток було вирішено використовувати плату SEN-11345 SparkFun Geiger Counter, яка обладнана вбудованим мікроконтролером ATmega328. Плата містить перетворювач високої напруги і підключається до хост-системи через стандартний COM-порт.

Одним із найскладніших завдань при проектуванні мобільних систем екологічного моніторингу є вибір газових датчиків. Більшість з них споживають 200-250 мА струму, що неприйнятно для мобільного застосування. Успішним винятком є газовий сенсор MiCS-6814 від SGX Sensortech [8]. MiCS-6814 - надійний MEMS сенсор. Оскільки він обладнаний аналоговим виходом, для спрощення проектування було вирішено використовувати готовий модуль Grove - Multichannel Gas Sensor. Він побудований на базі мікроконтролера ATmega168PA, та має інтерфейс I²C з програмованим адресом.

Цей набір датчиків, якщо необхідно, може бути розширений у майбутньому. Для спрощення підключення додаткових модулів доцільно використовувати «розумні» сенсори, які обладнані вбудованими засобами попередньої обробки та калібрування, з мікропроцесорним управлінням.

Таблиця 1. Система дозволяє визначати наступні параметри:

Фізіологічні	Навколишнього середовища
- Температуру тіла;	- Температуру повітря
- ЕКГ;	- Атмосферний тиск
- Частоту серцевих скорочень;	- Вологість
- Рівень кисню в крові;	- Рівень радіації (α , β , γ)
- Опір шкіри;	- Рівень ультрафіолетового випромінювання
- Частоту та глибину подиху.	- Рівень шкідливих газів *

* - Окис вуглецю CO (1 – 1000 ppm), двоокис азоту NO₂ (0,05 – 10 ppm), етанол C₂H₅OH (10 – 500 ppm), водень H₂ (1 – 1000 ppm), аміак NH₃ (1 - 500 ppm), метан CH₄ (>1000 ppm), пропан C₃H₈ (> 1000 ppm), 3-бутан C₄H₁₀ (>1000 ppm). ppm - parts per million – одна мільйонна частина або 10⁻⁶.

Окуляри доповненої реальності

Для того, щоб не відволікати увагу людини, яка використовує дану систему від виконання службових обов'язків, та своєчасно попереджати його про небезпеки, виявлені датчиками, дуже доцільно виводити отриману інформацію в поле зору оператора, накладаючи його на зображення оточуючої реальності. Для цього було вирішено використати окуляри доповненої реальності Intel/Recon Jet Pro [9].



Рисунок. 4. Окуляри доповненої реальності Intel/Recon Jet Pro.

Окуляри Recon обладнані двох ядерним ARM процесором частотою 1 ГГц, 1 Гб SDRAM, 8 Гб флеш-пам'яті, графічним чіпом, має величезну кількість вбудованих датчиків (3D акселерометр, 3D гіроскоп, 3D компас, барометр) і засобів зв'язку (GPS, Wi-Fi (IEEE802.11a/b/g/n 2.4GHz), Bluetooth 4.0 + BLE, ANT+TM). Мініатюрний екран, з роздільною здатністю 428x240, на який проектується зображення непрозорий, але його розміщення не заважає огляду. Екран

розташований в правому нижньому кутку периферичного зору. Екран включається тільки за бажанням людини - для цього служить спеціальна система виявлення погляду. Recon Jet обладнаний камерою з роздільною здатністю 1280x720 для зйомки відео і фото. Його вага становить 60 грам.

Висновки

Система персональної телеметрії дозволяє в реальному часі аналізувати отримані від первинних датчиків дані за рахунок вбудованого мікро контролера, і при виникненні критичних фізіологічних показників, передавати їх медикам швидкої допомоги, дільничному лікарю або в центр стеження за станом здоров'я співробітників служб порятунку. Аналіз комбінації показників різних датчиків дає додаткову інформацію. Наприклад, різке зменшення атмосферного тиску разом з даними акселерометра сповіщає про падіння людини (з точністю краще за 20 см), а підвищення тиску (і прискорення), про можливий вибух. Пристрій може бути доповнений або ж спрощений за рахунок видалення зайвих датчиків в залежності від передбачуваного використання. Розроблено прототип з мінімально необхідною кількістю сенсорів з дистанційним доступом по WEB-інтерфейсу. Масове використання аналогічних систем дозволить суттєво покращити показники виживання при інфаркті міокарда, інсульті, виникненні надзвичайних станів і т. д.



Рисунок. 5. Поле зору пожежника з показниками сенсорів які накладаються на зображення реального світу

Для підвищення автономності телеметричної системи доцільно використовувати сучасні альтернативні джерела живлення, які дозволяють перетворювати рух та деформацію людського тіла в електричний струм на основі п'єзоелектричних [10] або трибоелектричних [11] перетворювачів.

Подяка

Практична реалізація даного проекту стала можливою за рахунок гранту наданою Фондом Підтримки Людських Ресурсів Угорщини по програмі Підтримки Молодих Талантів Нації. Модуль DE10-Nano Kit наданий в якості підтримки компанією Terasic для участі в конкурсі Innovate FPGA design contest 2018 яка проводиться компанією Intel.



1. Buyya, R., Dastjerdi, A.V. (2016) Internet of Things: Principles and Paradigms. Morgan Kaufmann. – 378p.
2. MAXREFDES100#: Health Sensor Platform [Електронний ресурс]. – Режим доступу: https://www.maximintegrated.com/en/design/reference-design-center/system-board/6312.html/tb_tab0.
3. DE10-Nano User Manual, Terasic Inc., 2017. – 118p. [Електронний ресурс]. – Режим доступу: www.terasic.com.
4. RFS User Manual, Terasic Inc., 2017. – 62p. [Електронний ресурс]. – Режим доступу: www.terasic.com.
5. MLX90614 family Single and Dual Zone Infra Red Thermometer, Melexis, 2015. – 52p. [Електронний ресурс]. – Режим доступу: www.melexis.com.
6. BME280 Combined humidity and pressure sensor, 2015 [Електронний ресурс]. – Режим доступу: https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280_DS001-11.pdf
7. UVA Light Sensor with I²C Interface [Електронний ресурс]. – Режим доступу: <https://www.vishay.com/docs/84277/veml6070.pdf>
8. MiCS-6814 Data Sheet 1143 rev 8 [Електронний ресурс]. – Режим доступу: https://www.sgxsensortech.com/content/uploads/2015/02/1143_Datasheet-MiCS-6814-rev-8.pdf
9. Recon Jet™ Pro [Електронний ресурс]. – Режим доступу: <https://www.reconinstruments.com/enterprise/jet-pro/>
10. Molnar, A.A., Kuritnik I.P., Gerasimov V.V., Karabekov D.Zh. Piezoelectricity as a power source for portable electronic devices in the man-clothes system in Russian // Bulletin of University of Karaganda, Series Physics, 2015. – № 4 (80). – С. 62-65.
11. Molnar O., Gerasimov V., Kurytnik I.P. Triboelectricity and construction of power generators based on it // Przegląd Elektrotechniczny, 2018. – № 1. – PP. 167-171.

УДК 681.515.8

Поліщук М.М., Гринюк С.В., Білоус М.В.
Луцький національний технічний університет

ЛАБОРАТОРНИЙ БЛОК ЖИВЛЕННЯ НА БАЗІ МІКРОКОНТРОЛЕРА STM32F103VET6

М.М. Поліщук, С.В. Гринюк, М.В. Білоус. Лабораторний блок живлення на базі мікроконтролера STM32F103VET6. У статті представлено лабораторний блок живлення на базі мікроконтролера STM32F103VET6, його опис роботи та налаштування при складанні. Він може бути використаний для лабораторних досліджень та в навчальному процесі при вивченні комп'ютерних систем, комп'ютерної електроніки та схемотехніки.

Ключові слова: блок живлення, мікроконтролер STM32F103VET6, регулятор, енкодер, силова частина, цифрова частина, ARM Cortex M-3

Н.Н. Полищук, С. В. Гринюк, М.В. Билоус. Лабораторный блок питания на базе микроконтроллера STM32F103VET6. В статье представлены лабораторный блок питания на базе микроконтроллера STM32F103VET6, его описание работы и настройки при сборке. Он может быть использован для лабораторных исследований и в учебном процессе при изучении компьютерных систем, компьютерной электроники и схемотехники.

Ключевые слова: блок питания, микроконтроллер STM32F103VET6, регулятор, энкодер, силовая часть, цифровая часть, ARM Cortex M-3

M.M. Polishchuk, S.V. Grinyuk, M.V. Bilous. The laboratory block is on the basis of the microcontroller STM32F103VET6. The article presents a laboratory power supply based on the STM32F103VET6 microcontroller, its description of work and setup during assembly. It can be used for laboratory research and teaching in the study of computer systems, computer electronics and circuitry.

Key words: zhivlenya block, microcontroller STM32F103VET6, regulator, encoder, power part, digital part, ARM Cortex M-3

Вступ. На сьогоднішній день одним з найбільш потужних і перспективних засобів розробки електронних схем є мікроконтролери. Кожен розробник електронних пристроїв, а також ремонтники електроапаратури стикаються з проблемою джерела живлення для перевірки і ремонту створеного ними пристрою. Недорогі лабораторні джерела, які є в продажі, не завжди відповідають поставленим завданням, або занадто дорогі, та до того ж велика кількість органів управління не дозволяє оперативно змінювати характеристики, що подаються до приладу: напруги і струму. Пошук відповідних конструкцій, які б задовольняли всім вимогам, на просторах інтернету і публікаціях в журналах не принесла результату, або вони були малопотужні, або вкрай незручні. Тому існує потреба для розробки лабораторного блока живлення на базі мікроконтролера [1].

Під час роботи з мікроконтролерами необхідно мати на увазі таку обставину – коли розробляється система на основі мікроконтролера, то створюються не тільки апаратні засоби, що реалізуються відповідним підключенням мікроконтролера до зовнішніх пристроїв. Окрім цього розробник повинен забезпечити виконання багатьох системних функцій, які в традиційних мікропроцесорних системах забезпечуються за допомогою операційної системи та спеціальних периферійних мікросхем. Це, з одного боку, дещо ускладнює задачу, з іншого ж боку, дозволяє суттєво оптимізувати проект - як його апаратну, так і програмну частину для конкретного застосування. Відмінності в архітектурі процесорів можуть істотно позначитися на їхній продуктивності при виконанні різних завдань. [2]

Мета роботи – розробка лабораторного блока живлення на базі мікроконтролера STM32F103VET6, використання для лабораторних досліджень та в навчальному процесі при вивченні комп'ютерних систем, комп'ютерної електроніки та схемотехніки.

Аналіз існуючих блоків живлення. Сучасні електронні пристрої можна розділити на два типи: мобільні та стаціонарні. Мобільні використовують первинні джерела живлення, які мають запас електроенергії. Стаціонарні пристрої одержують живлення від вторинних джерел, які не виробляють власної енергії, а лише перетворюють електричний струм до необхідних параметрів (мережевої напруги 220 В).

На сьогоднішній день апаратура промислового виготовлення має в основному імпульсні джерела живлення. Основна ідея полягає в тому, що випрямлена напруга мережі перетворюється інвертором в змінну частотою в кілька десятків або сотень кілогерц. На таких частотах трансформатори виходять дуже малих розмірів, що дозволяє значно зменшити габарити і масу блоків живлення. Після трансформатора імпульсні напруги випрямляються і згладжуються фільтрами, що працюють на частоті мережі. Стабілізація вихідної напруги здійснюється в

© Поліщук М.М., Гринюк С.В., Білоус М.В.

первинної ланцюга за допомогою широтно-імпульсної модуляції – ШІМ, що також сприяє підвищенню ККД і зменшення габаритів блоку живлення.[3]

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Для розробки роботоздатного та якісного приладу, були сформовані наступні вимоги: напруга від 0 до 12 В; максимальний струм навантаження до 5А; зручність експлуатації та регулювання; можливість використання мікроконтролерного модуля в інших конструкція, шляхом зміни його конфігурації в сервісному режимі.

Спочатку було зпроектовано блок-схему пристрою (рис.1), яка складається з чотирьох блоків:

1. Первинне джерело призначене для гальванічної розв'язки від мережі живлення і зниження напруги для регулятора. Залежно від застосованого первинного джерела (імпульсний блок або трансформатор з секційної вторинною обмоткою) використовуються різні підпрограми мікроконтролера.

2. Регулятор – основна силова аналогова частина, яка здійснює регулювання напруги і струму в залежності від параметрів встановлених мікроконтролером, а також забезпечує компенсацію падіння напруги на струмовимірювальні резистори.

3. Мікроконтролер - забезпечує управління всією цією системою, збір даних про поточні значення - напруги на виході блоку, споживаний струм навантаженням. Індикація напруги, струму і поточний стан блоку, установка конфігурації, індикація перевищення струму навантаження понад встановлені значень. Запам'ятовування останніх встановлених параметрів.

4. Управління та індикація – це те що ми бачимо і те чим ми управляємо. Управління проводиться лише двома органами управління, це енкадер з кнопкою, і просто кнопка.

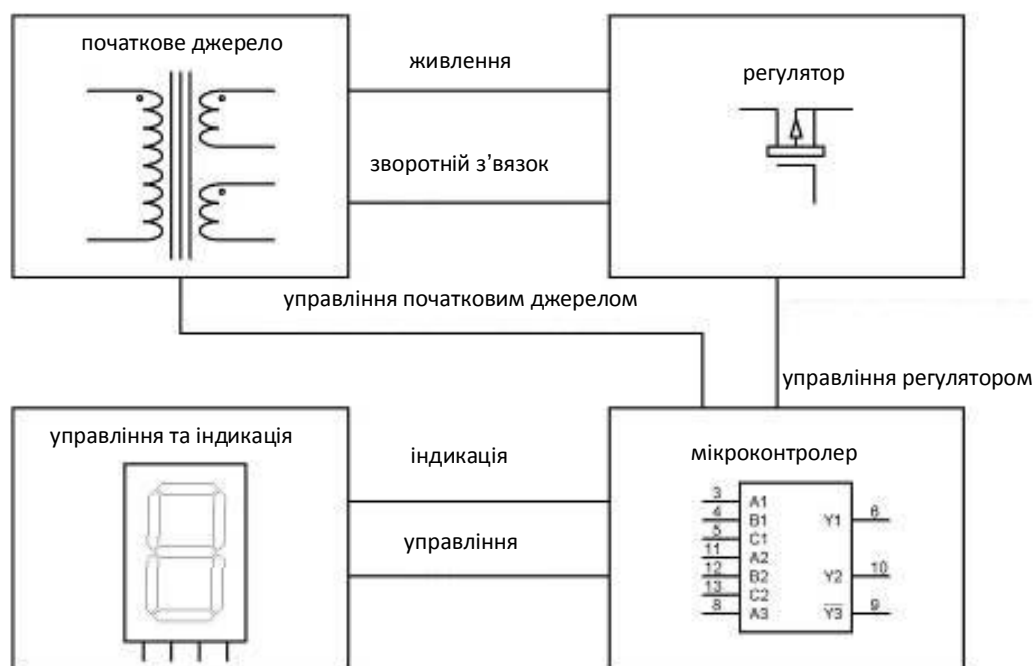


Рис.1 – Блок-схема лабораторного блока живлення

Виходячи з поставлених завдань, вибір припав на мікроконтролер STM32F103. Плата розробника у мінімальній конфігурації з мікроконтролером ARM Cortex M-3 STM32F103C8T6 повністю задовільняє всі вимоги. На платі зібрані всі необхідні елементи для початку роботи з даними сімейством мікроконтролерів (рис. 2). Для програмування контролера необхідно застосування програматора ST-Link.

Зручне розташування виходів дозволяє застосовувати її з безпечними макетними платами і програмувати не виймаючи її з макетки [4].

Характеристики:

- Модель: STM32F103C8T6.
- Ядро: ARM 32 Cortex-M3.
- Режим налагодження: swd.
- Робоча частота: 72 МГц.
- Пам'ять програм: 64 КБ
- Пам'ять даних: 20 КБ
- Напруга живлення: 2.0-3.6 В
- Кварцовий резонатор: 8 МГц і 32768 кГц
- Роз'єми: міні USB для живлення плати і для зв'язку з комп'ютером.
- Розмір: 5.3 см х 2.2 см.

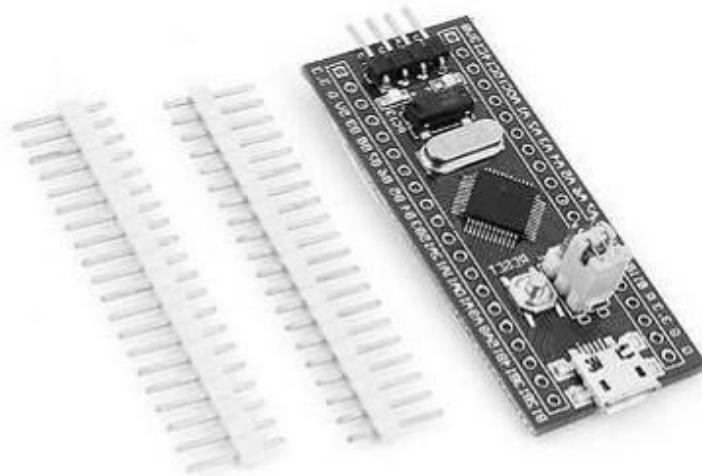


Рис. 2 – Плата з мікроконтролером STM32F103C8T6

Від більшості блоків живлення, описуваних в інтернеті [3], наш відрізняється методом зниження напруги і компактною формою корпусу. Насправді, це не повноцінний блок живлення, а понижуючий перетворювач напруги. Як DC-DC використовується перетворювач на мікросхемі LM2576, керований мікроконтролером. У блоках живлення з мікроконтролерним управлінням, конструкції яких зазвичай описуються в інтернеті, зниження напруги зазвичай проводиться лінійним методом – вся зайва потужність в таких блоках живлення розсіюється радіатором потужного транзистора.

Використання перетворювача DC-DC дозволяє відмовитися від використання великого радіатора. У разі, якщо навантаження не вимагає високої напруги, то для забезпечення великого струму можна використовувати слабкострумовий первинний блок живлення, але здатний видавати більш високу напругу (зазвичай я використовую первинний блок живлення 24 В, 0.8 А). Недолік же DC-DC - відносно високий рівень шумів по харчуванню (це важливо при роботі зі слабкими сигналами).

Мікроконтролерне управління потрібне для того, щоб напругу на виході такого блоку живлення можна було б точно встановлювати енкадером, при цьому регулювання напруги можна тимчасово заблокувати (щоб випадково не збити напругу, і не спалити пристрій, зачепивши ручку енкадера). Дискретність установки напруги можна змінювати. Можна керувати напругою, в залежності від струму (для заряду акумуляторів).

Характеристики отриманого блоку живлення:

- напруга живлення: 7-35 В;
- вихідна напруга: 1.3 - 30 В;
- максимальний струм: 3 А;
- дискретність установки напруги: 0.1 В;
- дискретність відображення струму: 0.01А (в блоці живлення немає стабілізації струму);
- захист від КЗ.

Блок живлення розбитий на дві частини (силову і цифрову), які зроблені на окремих платах. Схема блоку живлення, силова частина (рис 3).

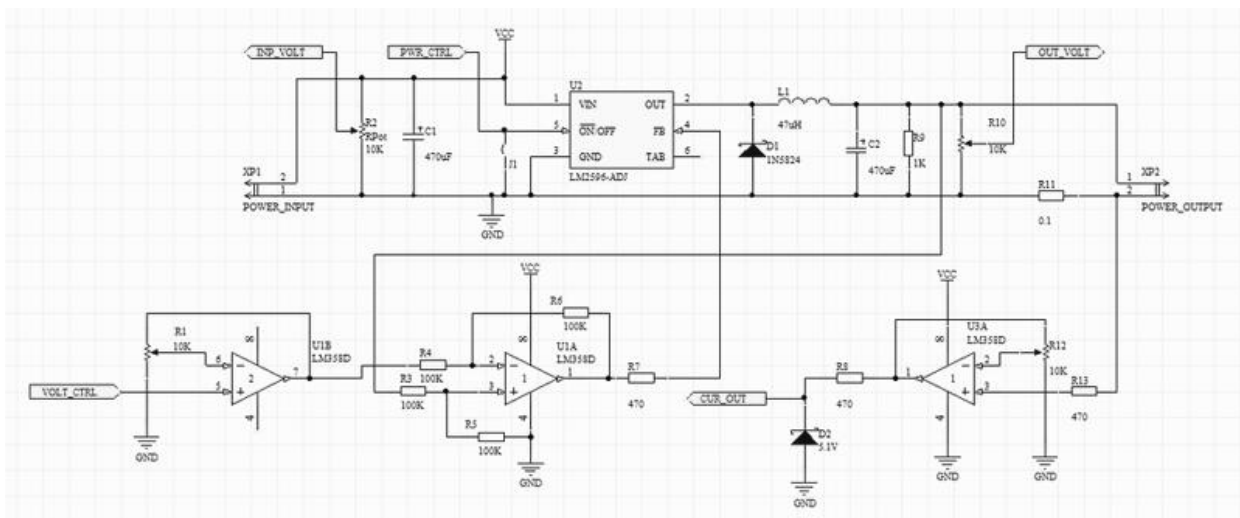


Рис. 3 – Схема блоку живлення (силова частина)

Обв'язка LM2576 або LM2596 стандартна – з даташіта. Дросель L1 в даному блоці живлення взятий з блоку живлення принтера (там був DC-DC), маркування на ньому не було. Взагалі, мікросхема LM2576 невимоглива до дроселя. Параметри дроселів для конкретних струмів і напруг дані в даташіте. Резистор R9 використовується для швидкого розряду конденсатора при відключенні напруги.

Для управління напругою з мікроконтролера використовуються ОУ U1. U1B підвищує напругу (3 В > 30В), U1A замикає зворотний зв'язок DC-DC, і дозволяє регулювати напругу на виході.

На ОУ U3 зібраний вузол вимірювання струму. Стабілітрон D2 захищає мікроконтролер від стрибків напруги, що викликаються стрибками струму (наприклад, при КЗ під час розряду конденсатора). Схема цифрової частини (рис. 4).

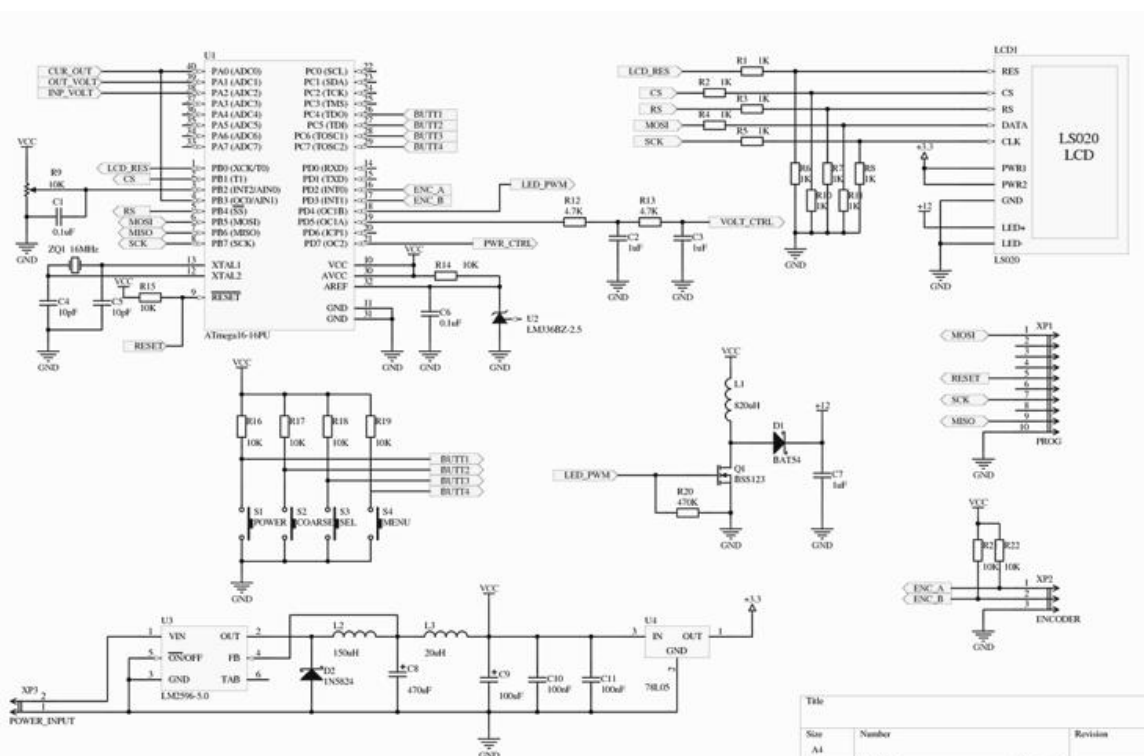


Рис. 4 – Схема цифрової частини

Живлення цифрової частини (5В) забезпечується від окремого DC-DC – так як вхідна напруга може бути великим, то звичайні лінійні стабілізатори можуть грітися, а місця під радіатор в корпусі немає. Увага - лінії VCC у цифровій і силовій частини різні.

Напруга для управління DC-DC формується за допомогою ШІМ, і фільтрується ФНЧ на R12, R13, C2, C3. Транзистор Q1 і його обв'язка служать для формування напруги 12 В для підсвічування індикатора (підвищує перетворювач). Резистор R9 задає струм спрацьовування захисту від КЗ (використовується компаратор контролера).

Кнопки, індикатор, енкодер встановлюються в передній панелі. Для захисту від брязкоту виходи енкодера з'єднуються через конденсатори 0,01 мкФ з землею.

Опис роботи з блоком живлення.

При включенні блоку живлення, на індикаторі кілька секунд відображається величина вхідної напруги і номер прошивки.

На дисплеї відображається напруга, яка встановлюється енкодером, та напруга на виході блоку живлення, виміряна АЦП (необхідне і бажане напруги можуть трохи відрізнитися при великому струмі або високих напруженнях).

Для вимірювання коефіцієнтів до блоку живлення через амперметр потрібно підключити навантаження, здатну витримати струм до 1 А. У нас використано досить потужну автомобільну лампочку.

Захист від КЗ – спрацьовує у всіх режимах по перериванню від вбудованого в контролер компаратора, при цьому подача харчування на навантаження відключається, на екран виводиться повідомлення. Через 0.5 сек проводиться перевірка - на навантаження подається напруга 1.3В, якщо струм при цьому перевищить 3А, то захист відключається, інакше процес повторюється. Фото пристрою зображено на рис. 5.

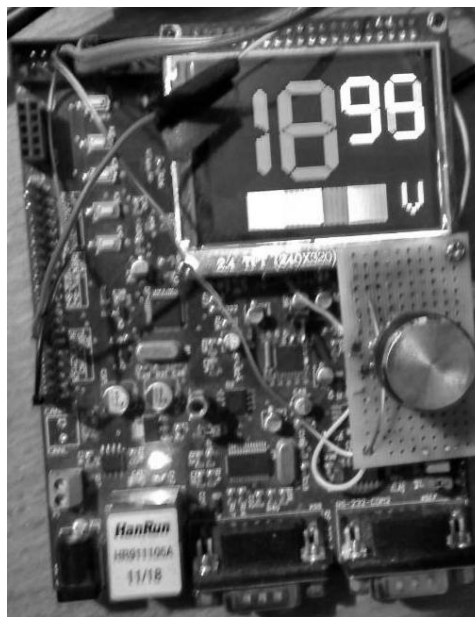


Рис. 5 – Лабораторний блок живлення на базі мікроконтролера STM32F103VET6

Налаштування блоку живлення при складанні.

Так як конструкція складається з двох частин, то і збиралася вона послідовно. Спочатку збирається силова частина. Після складання резистори R2, R10 встановлюються в нижнє становище. Це забезпечить захист контролера від перенапруги при наступному підключенні. Після установки перемички J1 і подачі напруги на вхід силовій частини, перевіряють її працездатність - на виході DC-DC має бути напруга не менше 1.3 В, яке повинно змінюватися при подачі зовнішнього напруги на лінію VOLT_CTRL. DC-DC повинен забезпечувати потрібний струм.

Потім збирається цифрова частина. Налаштування вона не вимагає (можливо, буде потрібно поміняти висновки енкодера місцями). Спочатку налаштовується індикація вхідної напруги

(резистором R2). Для контролю правильності настройки доведеться вмикати або вимикати первинний блок живлення. Подальша настройка йде в точному режимі.

Далі налаштовується коефіцієнт посилення ОУ, що відповідає за установку напруги. Енкодером потрібно встановити потрібне напруження, наприклад 10 В, підключити до виходу блоку живлення мультиметр, і повертаючи резистор R1, домогтися збігу напруг на екрані (бажаного) і мультиметри. Після цього, поворотом резистора R10 добиваються збігу напруг на екрані (дійсного) і мультиметри.

Після цього до виходу блоку живлення підключають навантаження і амперметр, енкодером встановлюють таку напругу, при якому струм в навантаженні близький, наприклад, до 1 А, і поворотом резистора R12 встановлюють таке ж значення струму на екрані. Після цього потрібно провести визначення коефіцієнтів струму, як описано вище.

Резистор R9 на цифровому блоці використовується для установки струму спрацьовування захисту від КЗ. Після підключення до блоку живлення навантаження, здатної витримати струм 3 А, і установки потрібного напруги, підлаштовують резистор, домагаючись спрацьовування захисту.

У разі одиночної плати при настройці приладу важливо контролювати положення резисторів R2, R10, щоб напруги на їх виходах не перевищували 5 В [5].

Висновки. Здійснено реалізацію лабораторного блока живлення на базі мікроконтролера STM32F103VET6, який може бути використаний для лабораторних досліджень та в навчальному процесі при вивченні комп'ютерних систем, комп'ютерної електроніки та схемотехніки.

1. Лабораторний блок живлення з управлінням від мікроконтролера [Електронний ресурс] – Режим доступу до ресурсу: <http://avr.ru/ready/contr/power/power> (дата звернення 23.03.2018)
2. Розробка радіоелектронних схем на основі мікроконтролерів (на прикладі AVR мікроконтролерів фірми Atmel): методичний посібник до курсу "Проектування радіоелектронних схем" для студентів радіофізичного факультету / Пархоменко Д. А., Смирнов Є. М. – Київ: Радіофізичний факультет Київського національного університету імені Тараса Шевченка, 2013. – 74 с.
3. Аладьшин Б. Блоки живлення електронних пристроїв [Електронний ресурс] – Режим доступу до ресурсу: <http://www.elektbruk.info/main/praktika/643-bloki-pitaniya-elektronnyh-ustroystv.html> (дата звернення 25.03.2018)
4. Плата разработчика STM32F103C8T6 ARM STM32 Минимальная конфигурация [Електронний ресурс] – Режим доступу до ресурсу: <https://arduino-ua.com/prod1328-plata-razrabotchika-stm32f103c8t6-arm-stm32-minimalnaya-konfiguraciya>. – Назва з екрану (Дата звернення 12.03.2018).
5. Незвичайний блок живлення на мікроконтролері [Електронний ресурс] – Режим доступу до ресурсу: <http://we.easyelectronics.ru/citizen/neobychnyy-blok-pitaniya-na-mikrokontrollere.html> (дата звернення 25.03.2018).
6. DC-DC Step Down модуль з заявленим струмом в 10 Ампер [Електронний ресурс] – Режим доступу до ресурсу: <https://mysku.ru/blog/china-stores/38587.html> (дата звернення 26.03.2018).
7. Основні схеми включення операційних підсилювачів [Електронний ресурс] – Режим доступу до ресурсу: http://stud.com.ua/28265/tovarovnavstvo/osnovni_shemi_vklyuchennya_operatsiynih_pidilyuvachiv (дата звернення 26.03.2018)
8. ЦАП в STM32 [Електронний ресурс] – Режим доступу до ресурсу: <http://easystm32.ru/for-beginners/37-dac-stm32> (дата звернення 01.04.2018).
9. STM32. Програмування STM32F103. ADC [Електронний ресурс] – Режим доступу до ресурсу: <http://www.avislab.com/blog/stm32-adc/> (дата звернення 01.04.2018).
10. STM32: Урок 6.2 - Таймери загального призначення і просунуті [Електронний ресурс] – Режим доступу до ресурсу: <http://robocraft.ru/blog/ARM/739.html> (дата звернення 06.04.2018).
11. STM32 і дисплей. Використання FSMC. [Електронний ресурс] – Режим доступу до ресурсу: <http://microtechnics.ru/stm32-i-displej-ispolzovanie-fsmc/> (дата звернення 06.04.2018).

УДК 378.147 (045)
Поплавська Г.В., викладач
Луцький педагогічний коледж

НАПРЯМКИ ВПРОВАДЖЕННЯ НОВИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У ПРОЦЕСІ КОНТРОЛЮ ТА ОЦІНКИ ЗНАНЬ СТУДЕНТІВ

Поплавська Г.В. Напрямки впровадження нових інформаційних технологій у процесі контролю та оцінки знань студентів. У статті розглянуто основні напрямки впровадження нових інформаційних технологій у процесі контролю та оцінки знань студентів.

Ключові слова: нові інформаційні технології(НІТ), комп'ютерні технології навчання(КТН), навчальний процес, тести.

Поплавская А. В. Направления внедрения новых информационных технологий в процессе контроля и оценки знаний студентов. В статье рассмотрены основные направления внедрения новых информационных технологий в процессе контроля и оценки знаний студентов.

Ключевые слова: новые информационные технологии (НИТ), компьютерные технологии обучения (КТС), учебный процесс, тесты.

Poplavskaya G.V .Directions of introduction of new information technologies in the process of control and evaluation of students' knowledge. In the article the main directions of introduction of new information technologies in the process of control and evaluation of students' knowledge are considered.

Key words: new information technologies (NIT), computer learning technologies (CLT), educational process, tests.

Постановка наукової проблеми.

Тенденції розвитку сучасного суспільства, його спрямованість на інформатизацію всіх галузей виробництва, суспільного і особистого життя, визначають необхідність все більш широкого використання інформаційних технологій в освітніх областях.

Інформатизація (у перекладі з лат. - пояснення, виклад) є одним з головних напрямків сучасної науково-технічної революції, на якому ґрунтується перехід від індустріального етапу розвитку суспільства до інформаційного. Це процес перебудови життя суспільства на основі використання достовірного, вичерпного і своєчасного знання в усіх суспільно значущих видах діяльності.

Нові перспективи інформатизації освіти відкриває Закон України від 9.01.2007 р. № 537 - V «Про основні засади створення інформаційного суспільства в Україні на 2007- 2015 роки», де визначені такі завдання: прискорення розробки та впровадження новітніх конкурентоспроможних інформаційно-комунікаційних технологій (ІКТ) в усі сфери суспільного життя забезпечення комп'ютерної та інформаційної грамотності населення, насамперед шляхом створення системи освіти, орієнтованої на використання новітніх ІКТ у формуванні всебічно розвиненої особистості, створення загальнодержавних інформаційних систем у сфері освіти і науки [1, с. 210-214].

Інформатизація освіти — упорядкована сукупність взаємопов'язаних організаційно-правових, соціально-економічних, науково-методичних, науково-технічних, виробничих та управлінських процесів, спрямованих на задоволення інформаційних, обчислювальних і телекомунікаційних потреб, пов'язаних з можливостями методів і засобів інформаційних та комунікаційних технологій учасників навчально-виховного процесу, а також тих, хто цим процесом керує і його забезпечує [6, с. 360]. Отже, інформатизація освіти розглядається сьогодні в суспільстві як першочергова проблема.

Ця проблема визначає мету дослідження.

Аналіз досліджень і публікацій . Фундаментальні дослідження в галузі інформатизації освіти робили В.Ю. Биков, М.І. Жалдак, І.А. Король, М.М. Фіцула та багато інших. На основі аналізу психолого-педагогічних досліджень (М.І. Жалдак, В. Глушков, В.Ю. Биков, А.С. Гринберг, І.А. Король та ін.) зроблено висновок, що «нові інформаційні технології» (скорочено «НІТ») визначаються як сукупність методів і технічних засобів збору, організації, зберігання, обробки, передачі та подання інформації, що поглиблює знання людей і розвиває їхні можливості з управління технічними та соціальними проблемами. Інформаційна технологія являє собою організовану сукупність інформаційних процесів з використанням обчислювальної техніки, що забезпечує оперативну обробку даних, швидкий пошук необхідної інформації, доступ до різних джерел інформації.

Останнім часом інформаційні технології використовуються у вищій школі при проведенні різних видів контролю на основі спеціально розроблених комп'ютерних програм тестування. М.М. Фіцула зазначає, що такий вид контролю дозволяє ефективно використовувати час, ставить перед усіма студентами однакові вимоги, допомагає уникати надмірних хвилювань.

Формулювання цілей статті. Мета статті – розглянути основні напрямки впровадження НІТ у процесі контролю та оцінки знань студентів, дослідити досягнення та результативність через тестування, яке спрямоване на оцінку набутого рівня розвитку здібностей, навичок і знань студентів.

Основні результати досліджень. Інформатизація охоплює три взаємопов'язані процеси:

- медіатизації (у перекладі з лат. - посередник) – вдосконалення засобів збору, зберігання і розповсюдження інформації;
- комп'ютеризації – вдосконалення засобів пошуку та обробки інформації;
- інтелектуалізації – розвиток здібностей, сприйняття та продукування інформації, тобто підвищення інтелектуального потенціалу суспільства, в т.ч. використання засобів штучного інтелекту.

У широкому сенсі «інформаційні технології» – це комплекс соціально-педагогічних перетворень, пов'язаних з насиченням освітніх систем інформаційною продукцією, засобами і технологією, у вузькому – впровадження у заклади системи освіти інформаційних засобів, які ґрунтуються на використанні комп'ютерної техніки.

Одним з найбільш істотних складових інформатизації вищих навчальних закладів є інформатизація навчального процесу, яка передбачає:

- оперативне оновлення навчальної інформації у зв'язку з розвитком науки, техніки, культури;
- організації процесу навчання з урахуванням індивідуальних особливостей кожного студента;
- використання диференційованого підходу використання автоматизованого навчання для вирішення різних завдань;
- обробки результатів вимірювань експериментальних досліджень проведення контролю підготовленості студентів.

Все це сприяє підвищенню інформаційної культури студентів, здійсненню диференціації навчально-виховного процесу з метою розвитку нахилів та здібностей студентів, задоволення їх запитів і потреб, розкриття творчого потенціалу, удосконалення управління освітою, підвищенню ефективності наукових досліджень.

Нові інформаційні технології припускають використання усього розмаїття сучасних пристроїв обробки інформації та їх периферійного обладнання [7, с. 192].

Для інформаційного забезпечення педагогічного процесу кожен викладач і студент повинні мати доступ до практично необмеженого обсягу інформації та її аналітичної обробки.

Комп'ютерна (інформаційна) технологія навчання (КТН) спрямована на досягнення мети інформатизації навчання шляхом застосування комплексу функціонально залежних педагогічних, інформаційних, методологічних, психофізіологічних, ергономічних засобів і методик, створених і організованих на базі технічного та комп'ютерного програмного забезпечення.

Важливою особливістю комп'ютерних програм є їх висока інформаційна насиченість, що відкриває додаткові можливості для раціонального використання навчального часу, підвищення ефективності роботи викладача.

Програми, які використовуються у вищих навчальних закладах, поділяють на:

- навчальні (скеровують навчання, враховуючи наявні знання та індивідуальні здібності студентів, а також сприяють засвоєнню нової інформації);
- діагностичні (тестові) (призначені для перевірки, оцінки знань, умінь, здібностей);
- тренувальні (розраховані на повторення пройденого навчального матеріалу);
- імітаційні (створюють певний аспект реальності за допомогою параметрів для вивчення його основних структурних або функціональних характеристик);
- моделюючі (відображають основні елементи і типи функцій, моделюють певну реальність);
- програми типу «мікросвіт» (створюють віртуальне навчання);
- інструментальні програмні засоби (забезпечують виконання конкретних операцій: обробки тексту, складання таблиць, редагування графічної інформації).

Використання КТН у сучасному навчанні має певні переваги над традиційними засобами навчання (необмежені можливості комп'ютерної обробки інформації, посилення мотивації навчання, активну участь у навчальному процесі, самостійна перевірка рівня власних знань, гнучкість видів контролю тощо).

Використання ІТ дозволяє значно підвищити ефективність інформації за рахунок її своєчасності, доцільного дозування, доступності (зрозумілості), мінімізації, оперативного взаємозв'язку джерела навчальної інформації та студента, адаптації темпу подання навчальної інформації в швидкості її засвоєння, врахування індивідуальних особливостей студентів, ефективного поєднання індивідуального та колективного діяльності, методів і засобів навчання, організаційних форм навчального процесу, значною мірою сприяє вирішенню проблем його гуманізації.

На сучасному етапі соціальних і технологічних перетворень однією з вимог до всіх учасників навчального процесу у вищих навчальних закладах є готовність майбутнього вчителя до використання інформаційно-комунікаційних технологій, комп'ютеризованих систем в цілому у навчанні та професійній діяльності.

Виділяють три рівні такої готовності: сучасний рівень володіння комп'ютеризованими технологіями, вміння безпосередньо використовувати їх у професійній діяльності, вміння підвищувати свій професійний рівень за допомогою засобів КТН. Важливим елементом цієї готовності є не тільки теоретична підготовка у певній галузі знань, а також і практичні вміння викладача організувати і проводити навчальне заняття за допомогою комп'ютерних засобів і технологій; підготувати навчальне заняття за дистанційною формою навчання; застосувати фірмові розробки електронних навчальних посібників, створити власний електронний навчальний посібник з конкретної дисципліни.

Для ефективного використання у навчальному процесі сучасних інформаційно-комунікаційних технологій викладач повинен володіти певними специфічними вміннями (за М. Жалдаком):

- застосовувати сучасні інформаційно-комунікаційні технології у підготовці, аналізі, коригуванні навчального процесу, управлінні навчальним процесом і навчально-пізнавальною діяльністю студентів;
- підбирати раціональні методи і засоби навчання, враховувати індивідуальні особливості студентів, їх нахили та здібності;
- ефективно поєднувати традиційні методичні системи навчання з новими інформаційно-комунікаційними технологіями [4].

Впровадження у навчальний процес у вищій школі нових інформаційних технологій є об'єктивним процесом розвитку освіти. Однак вони не повинні використовуватися педагогами бездумно, оскільки одну з технологій не можна вважати універсальною: кожна з них у різних ситуаціях дає різні результати, і це необхідно враховувати при їх виборі.

Останнім часом інформаційні технології використовуються у вищій школі при проведенні різних видів контролю на основі спеціально розроблених комп'ютерних програм тестування. М.М. Фіцула зазначає, що такий вид контролю дозволяє ефективно використовувати час, ставить перед усіма студентами однакові вимоги, допомагає уникати надмірних хвилювань. Тестова перевірка виключає випадковість в оцінці знань, стимулює студентів до самоконтролю [7, с. 222].

За період свого існування тести зазнали помітну еволюцію, і сьогодні тестування є цілою галуззю. Тести використовуються у ряді областей науково-практичної діяльності людини. Істотну роль вони відіграють і в системі освіти [2, с. 31].

Комп'ютерне тестування як підсумкова форма контролю, яка використовується у вищій школі з метою удосконалення навчального процесу, вимагає детальної розробки етапності його проведення, тобто відповідної технології. Технологія комп'ютерного тестування розглядається нами як модель навчального процесу, має чітку послідовність дій, починаючи з розробки викладачем тестів, підготовки студентів до їх виконання, організації та проведення викладачем підсумкової форми контролю, оцінки та корекції знань за умови забезпечення комфортності і зворотного зв'язку між суб'єктами навчання.

У своєму дослідженні ми акцентували увагу на тестах досягнень та результативності, оскільки вони спрямовані на оцінку набутого рівня розвитку здібностей, навичок і знань студентів. Дехто вважає, що ці тести відображають не стільки вплив накопиченого досвіду на формування умінь тих чи інших завдань, скільки вимірюють вплив програми навчання,

професійної підготовки на ефективність засвоєння комплексу знань, формування різних спеціальних умінь і навичок [2, с. 69]. Тести досягнень та успішності мають перевагу над іншими видами оцінки. Їх показники орієнтовані на вимірювання ключових понять, тем та елементів програми, в той час як інші види оцінки, таких як контрольна робота, передбачають оцінку конкретної сукупності знань в детальному й послідовному викладі. З урахуванням виду контролю були використані тести підсумкового контролю.

Підвищення об'єктивності вимірювання результатів навчальної діяльності студентів сприяє застосуванням різних за конструкцією тестових завдань: відкритої форми (короткі однозначні відповіді вільної форми), закритої форми (вибір відповіді з певної кількості варіантів, вибір кількох правильних відповідей, вибір найбільш правильної відповіді), тестів-альтернатив, тестів-відповідностей і т.д.

При оцінці знань студентів за допомогою комп'ютерних тестів потрібно враховувати ту обставину, що деякі правильні відповіді можна дати шляхом відгадування. Найрадикальніший спосіб ліквідації наслідків вгадування варіантів відповідей — це використання завдань, де даний процес повністю виключений. Це завдання відкритої форми (на доповнення або з вільно конструйованою відповіддю) та закритої форми (на встановлення співвідношення, на встановлення правильної послідовності або завдань з вибором кількох правильних відповідей і т.д.).

Отже, перший етап у технологічному ланцюжку комп'ютерного тестування – розробка тестів викладачем. Основні труднощі для викладача вищої школи на даному етапі роботи становить складання необхідного «банку» тестів.

Зупинимося на технології розробки тестів, тому що вона складна і трудомістка.

При складанні тестів за основу бралися наступні положення:

- тести повинні охоплювати ключові питання теми, розділу або навчального курсу, відповідати робочій програмі з дисципліни;
- тести повинні бути конкретними і лаконічними;
- відповіді на тестові питання повинні бути чіткими, зрозумілими і однозначними;
- тестові завдання повинні бути різними з побудови та рівнем складності і оцінюватися відповідною шкалою;
- тести повинні бути стандартними, тобто придатними для широкого практичного використання.

Під час складання тестових завдань важливим є коректне використання так званих дистракторів (від англ. distract – відволікати увагу, збивати з пантелику) – варіанти відповідей у завданнях з вибором, які не є правильними, але зовні близькі до правильного рішення. Підбір адекватних дистракторів є невід'ємною частиною процедури розробки надійного і валідного тесту, складеного на матеріалі завдань з вибором. Саме цей елемент створює додаткову мотивацію для роботи з підручником. Пояснюється все дуже просто. Якщо тест складений якісно, то в ньому присутня достатня кількість сильних дистракторів. Їх наявність служить основою неправильних відповідей. Студент переконується, що ймовірність відгадування правильних відповідей дуже мала. Це викликає бажання взяти конспект лекції або підручник і розібратися, знайти правильні відповіді на питання тестових завдань.

З метою вдосконалення комп'ютерного тестування студентів, на нашу думку, особливо вдалим є використання тестів різних рівнів складності, що дозволяє більш об'єктивно оцінювати знання, вміння та навички студентів.

Перший рівень складності тестового контролю передбачає засвоєння понятійного апарату навчальної дисципліни, репродуктивне відтворення програмного матеріалу, другий рівень — розуміння основних законів, положень, закономірностей, відповідностей, які необхідні для засвоєння навчального курсу, третій рівень вимагає творчого підходу до виконання завдань практичного плану.

Таким чином, щоб тестовий контроль знань був результативним, необхідно дотримуватися таких психолого-педагогічних вимог щодо його застосування:

- важливим є дотримання організаційної чіткості у проведенні тестового контролю, яка передбачає: наявність організаційного моменту, під час якого викладач пояснює тестові завдання, дає відповіді на запитання студентів, обов'язково визначає час, необхідний для виконання роботи;
- тести підсумкового контролю повинні бути різного ступеня складності і оцінюватися

різною кількістю балів;

- тести підсумкового контролю не повинні бути доступними для студентів у передекзаменаційний період;
- обов'язково слід здійснювати аналіз результатів тестування з метою подальшого коригування навчального процесу.

Загалом, використання комп'ютерної тестової форми контролю забезпечує:

- високий рівень навчальних досягнень;
- виховує ціннісне ставлення до навчальної діяльності;
- розвиває пам'ять, логічне мислення;
- розвиває вміння робити правильний вибір;
- знижує рівень тривожності;
- допомагає вчителю здійснювати своєчасно корекцію знань.

При сучасному розвитку ринку програмного забезпечення та систем комп'ютерного тестування розроблено досить багато програм для комп'ютерного тестування знань студентів. Ці системи являють собою або окремий програмний комплекс, що вимагає установки на комп'ютер кінцевого користувача, або Інтернет-сайт, що дозволяє проводити процес тестування й аналіз його результатів за допомогою звичайних веб-браузерів.

У своїй практиці я використовую такі програми для створення комп'ютерних тестів: MyTest, TestYourSelf, MiniTestSL, Опіон, Конструктор тестів, MyTest Pro, Test-W і Test-W2, які легко встановлюються на комп'ютер і є досить функціональними [3].

Комп'ютерне тестування є одним з найоптимальніших засобів контролю, який задовольняє вимоги щодо об'єктивності отриманої оцінки, якості процесу контролю та має позитивний вплив на мотивацію, зацікавленість студента до процесу навчання в цілому. Але, незважаючи на вагомі переваги, воно має і свої недоліки (наприклад: наявність випадкових, несистематичних помилок вимірів; відсутність чітких математичних критеріїв оцінки, неякісні тестові матеріали).

Після тестування викладач отримує відомості щодо володіння навчальним матеріалом конкретного курсу кожним студентом, який пройшов тестування. Аналізуючи ці дані, можна чітко визначити закономірності проблем із засвоєнням навчального матеріалу по конкретні темі, але про причини цих труднощів та пропусків, викладач може й не знати. Не завжди питання комп'ютерного тестування зможуть перевірити творчий рівень знань студента. Часто студенту не вистачає часу, який виділяється на тестування, для більш детального аналізу питання чи для пригадування матеріалу.

Комп'ютерне тестування вимагає від викладача творчого підходу до складання тестових завдань для того, щоб можна було впевнитись в об'єктивності та достовірності отриманої оцінки студентом. Але, на жаль, є проблема відсутності підручників зорієнтованих на тестову форму контролю знань, на значні обсяги часу, необхідного для первинної підготовки якісних тестів.

У комп'ютерному тестуванні, як і в будь-якому процесі, не можна забувати про елемент випадковості. Причиною цього може бути, як некоректно поставлене питання, так і елементарне вгадування відповіді на питання. Як наслідок – спотворені результати тестування.

Створювати тест, що містить завдання з багатозначними питаннями і відповідями, виявляється, дуже складно. Звичайно, відомі технологічні розробки, присвячені застосуванню техніки для контролю знань, не вичерпують всіх можливостей комп'ютера. Можливості комп'ютера в сфері освіти нам ще належить відкривати і вивчати, запроваджуючи педагогічні інновації – введення в практику навчання новизни, що дозволяє більш ефективно вирішувати назріваючі проблеми.

Висновки та перспективи подальших досліджень. Як висновок, слід ще раз підкреслити: тест як система завдань специфічної форми і відповідного змісту є науково-обґрунтованим інструментом оцінки знань, умінь і навичок студентів, допомагає здійснювати індивідуальний контроль результатів навчання кожного з них, мобільно керувати навчально-виховним процесом. У той же час існує необхідність створення чіткої технології проведення даної форми контролю, пов'язаної з використанням відмінних тестів для поточного та підсумкового контролю, проведення на кафедрах педагогічної експертизи готовності тестових завдань для оцінки знань студентів, постійного вдосконалення тестових комп'ютерних програм. Адже,

розвиток обчислювальної техніки та мультимедіа дає змогу створювати навчальні та тестові комп'ютерні програми, що враховують індивідуальні особливості студентів, підвищуючи мотивацію та рівень засвоєння матеріалу. Навчальні заклади з традиційними формами системи вищої освіти, якщо вони хочуть вижити у ХХІ столітті, змушені внести серйозні зміни до своєї діяльності й, передусім, до пропозицій послуг та систем подання знань.

1. Гуржий А.Н. Приоритетные направления информатизации общеобразовательных учебных заведений // Педагогическая и психологическая наука в Украине: Сб. науч. трудов к 15-летию АПН Украины в 5 томах. / Т.5. Непрерывное профессиональное образование: Теория и практика. - К. : Педагогическая мысль, 2007. - с. 207-218.
2. Гуревич Р.С., Кадемія М.Ю. Інформаційно-комунікаційні технології у навчальному процесі: посібник для педагогічних працівників і студентів педагогічних вищих навчальних закладів. – Вінниця: ДОВ "Вінниця", 2002. –116с.
3. Гуревич Р.С. Компетентісна освіта у вищій педагогічній школі: методичний посібник для викладачів педагогічних вищих навчальних закладів / Р.С. Гуревич, М.Ю. Кадемія, Л.С. Шевченко –[2-ге вид., доповнене]. – Вінниця: ТОВ фірма «Планер», 2010.–166 с.
4. Жалдак М.И. Система подготовки учителя к использованию информационной технологии в учебном процессе: Дис. в форме науч. доклада д-ра пед. наук: 13.00.02. М., 1989. - 48 с.
5. Энциклопедия образования / Акад. пед. наук Украины, главная ред. В.Г. Кремень. - М.: Юридическая литература, 2008. - с. 360.
6. Кадемія М. Ю., Шахіна І.Ю. Інформаційно-комунікаційні технології в навчальному процесі : Навчальний посібник / М. Ю. Кадемія, І. Ю. Шахіна. – Вінниця : ТОВ «Планер», 2011.
7. Ляшова Н., Спивакова А. Оценка знаний студентов методом тестирования // Начальная школа. - 2000. - № 4. - с. 69-70.
8. Фицула Н.Н. Педагогика высшей школы: Учеб. пособие. - М.: Академвидав, 2006. - 352 с.

УДК 330.43

Ройко Л.Л. к.пед.н. доц., Ройко О.О. студентка

Східноєвропейський національний університет імені Лесі Українки

ПРИКЛАДНА СПРЯМОВАНІСТЬ КУРСУ “МАТЕМАТИКА ДЛЯ ЕКОНОМІСТІВ ТА ЕКОНОМІЧНЕ МОДЕЛЮВАННЯ”

Ройко Л.Л., Ройко О.О. Прикладна спрямованість курсу “Математика для економістів та економічне моделювання”. У статті обґрунтовано реалізацію прикладної спрямованості курсу “Математика для економістів та економічне моделювання” з використанням прикладних задач.

Ключові слова: прикладна задача, математика для економістів, економічне моделювання, студент економічного профілю.

Ройко Л.Л., Ройко О.О. Прикладная направленность курса “Математика для экономистов и экономическое моделирование”. В статье обосновано реализацию прикладной направленности курса “Математика для экономистов и экономическое моделирование” с использованием прикладных задач.

Ключевые слова: прикладная задача, математика для экономистов, экономическое моделирование, студент экономического профиля.

Royko L.L., Royko O.O. Applied orientation of the course “Mathematics for economists and economic modeling”. The article substantiates the implementation of the applied orientation of the course “Mathematics for economists and economic modeling” with the use of applied tasks.

Keywords: applied problem, mathematics for economists, economic modeling, economics student.

Постановка проблеми. Метою системи вищої освіти – є формування кваліфікованого та конкурентоспроможного фахівця, який володіє сучасними технологіями виробництва у своїй галузі, здатністю до прийняття нестандартних рішень, усвідомленням завдань і засобів самовдосконалення, розвиненими навичками самоосвіти. Факультет “Міжнародних відносин” у Східноєвропейському університеті імені Лесі Українки (СНУ) забезпечує підготовку фахівців з міжнародної економіки для різних підрозділів зовнішньоекономічної діяльності, приватного бізнесу, державних установ. Міжнародна економіка як спеціалізація є однією з провідних і постійно збільшує значущість міжнародної діяльності як основного чинника підвищення рівня конкурентоспроможності. Стрімкі зміни зовнішнього середовища, прискорення темпів науково-технічного прогресу, потреба у впровадженні нових технологій для забезпечення конкурентних переваг підприємств на внутрішньому та зовнішньому ринках вимагає нових підходів у забезпеченні міжнародних зв'язків.

Навчальна дисципліна “Математика для економістів та економічне моделювання” у (СНУ) передбачена для студентів підготовки бакалавр, спеціальності: 292 – міжнародні економічні відносини, за освітньою програмою: міжнародні економічні відносини. Дисципліна викладається у 4 семестрах і складає 270 годин.

Викладання у великому обсязі чистої математики – неефективна справа, яка не сприймається студентами. Математика стає чужою для них наукою, якщо вони не бачать у ній можливості використання у майбутній професійній діяльності. Тому слід кардинально змінити стиль викладання математики для економістів міжнародників, який передбачав би доступне, комплексне викладання класичних розділів вищої математики, теорії ймовірностей, математичної статистики, економіко-математичного моделювання, фінансової математики і реалізацію тісних зв'язків з економікою. У процесі реалізації цих зв'язків основні поняття і методи математики повинні бути підкріплені сучасними економічними поняттями і розв'язуванням актуальних завдань ринкової економіки.

Одним із важливих засобів реалізації прикладної спрямованості курсу “Математика для економістів та економічне моделювання”, на наш погляд, є використання прикладних задач, які відображають реальні економічні ситуації та процеси, а їх розв'язання показує необхідність застосування математичного апарату в економіці.

Аналіз наукових досліджень і публікацій. Методологічну базу дослідження склали ідеї: застосування методів математичного моделювання в економіці (Г. Берегова, О. Бобик, І. Буркінський, В. Вітлінський, Б. Грабовецький, В. Здрок, Н. Лепа, В. Осипов, С. Прокопов, К. Рум'янцева, Є. Слуцький та інші); проблеми розробки та впровадження активних методів навчання (В. Буркова, Г. Ковальчук, В. Петрук, І. Смолін та інші); різні аспекти підготовки

фахівців економічного профілю (Н. Ванжа, Г. Дутка, Н. Захарченко, Т. Коваль, Л. Нічуговська, Т. Поясок, О. Смілянець, Ю. Ткач та інші) [1].

Професійну спрямованість навчання вищої математики студентів економічного профілю розглянуто у дисертаційних дослідженнях В. Зінченко, Л. Гусак, А. Савіної, Н. Самарук, І. Коновалової, О. Попової, К. Словак та інших [5].

Питання прикладної спрямованості вищої математики з використанням прикладних задач відображено у працях Л. Васяк, К. Власенко, І. Гурана, О. Гутіка, І. Главатських, Т. Думанської, О. Королюка, В. Коваленко, Т. Крилової, Л. Нічуговської, В. Чернишева та інших.

У працях Т. Архіпової, М. Бакланової, О. Ващук, Є. Вінниченка, М. Голованя, С. Лешук, І. Іваськіва, С. Семерікова, О. Собаєвої, О. Співаковського показано значимість впровадження у навчальний процес ІКТ.

Дослідження В. Бикова, М. Жалдака, Т. Капустіної, В. Клочка, Ю. Лотюка, К. Словак, С. Ракова, Ю. Рамського, М. Рафальської, Ю. Триуса, С. Шокалюк показали, що одним з найбільш ефективних засобів ІКТ навчання математики є системи комп'ютерної математики (СКМ).

Мета статті – обґрунтувати реалізацію прикладної спрямованості курсу “Математика для економістів та економічне моделювання”, що викладається студентам міжнародникам, з використанням прикладних задач.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Застосування математики в економіці дає змогу формалізувати істотні зв'язки між економічними об'єктами, а використання математичної термінології – стисло і точно формулювати економічні проблеми. Сучасні підходи до розв'язання цих завдань передбачають наявність ґрунтовних знань та вмінь фахівців із математичного моделювання, математичних методів обробки інформації, оптимізації економічних задач, визначення економічних ризиків, застосування кількісних методів прийняття ефективних рішень тощо. Саме ці знання та вміння закладають основи успішного засвоєння дисциплін економічного циклу: міжнародного та фінансового менеджменту, міжнародних фінансів, маркетингу, проблем міжнародних стратегій економічного розвитку, практичних методів управління зовнішньоекономічною діяльністю, макроекономіки, мікроекономіки, статистики, економіки підприємств, економічного аналізу, економічного ризику, національної економіки, регіональної економіки, управління витратами тощо.

Більшість тем із дисциплін цього циклу містять матеріал, який можна ефективно використовувати для наочної ілюстрації та практичного використання при вивченні математичного матеріалу, що сприятиме глибокому й більш усвідомленому вивченню абстрактної математичної теорії, а також підвищенню інтересу студентів до вивчення математики.

Наведемо приклади міжпредметних зв'язків деяких розділів дисципліни “Математика для економістів та економічне моделювання” з економічними дисциплінами та основні задачі на їх використання [4]:

– **елементи лінійної алгебри:** макроекономіка, економічна теорія, економіка підприємств, економіко-математичне моделювання, оптимізаційні методи і моделі (задачі: модель багатогалузевої економіки Леонтьєва, лінійна модель обміну (модель міжнародної торгівлі)).

– **елементи векторної алгебри:** економіка підприємств, оптимізаційні методи і моделі (задачі: простір товарів, вектор цін).

– **елементи аналітичної геометрії:** економіка підприємств, мікроекономіка, оптимізаційні методи і моделі (задачі: лінійні моделі виробничих функцій, лінійні моделі попиту і пропозиції, аналіз прибутковості – збитковості на основі лінійних моделей функцій доходу і витрат, закон розподілу прибутків (закон Парето)).

– **функція однієї та багатьох змінних:** мікроекономіка, економічна теорія, статистика, економетрика, економіка підприємств (задачі: функції попиту і пропозиції, рівноважна ціна і павутиноподібна модель, виробничі функції: функція витрат, функція доходу, функція прибутку, функція собівартості, функція залежності попиту на різні товари від доходу населення, прості та складені відсотки, задача про неперервне нарахування відсотків, економічна інтерпретація числа e , функція Кобба-Дугласа).

– **диференціальне числення функції однієї та багатьох змінних:** оптимізаційні методи і моделі, мікроекономіка, економетрика, кількісні методи фінансового прогнозування, фінанси підприємств (задачі: економічний зміст похідної, похідна функції обсягу виробництва як продуктивність праці, похідна виробничої функції як: граничні витрати, граничний виторг, граничний дохід, граничний прибуток виробництва, еластичність функції однієї змінної і

частинні еластичності функції багатьох змінних виробничих функцій, функції попиту і пропозиції, максимізація доходу і прибутку та мінімізація витрат у випадку виробничих функцій однієї та багатьох змінних, мінімальність транспортних витрат, опуклість функції корисності та її економічний зміст, оптимізація оподаткування підприємств).

– **інтегральне числення:** теорія ймовірностей та математична статистика, мікроекономіка, фінанси підприємств, статистика (задачі: обчислення загальних витрат, доходу, прибутку за відомими відповідними граничними витратами, доходом, прибутком, обчислення обсягу виробленої продукції за відомою продуктивністю праці, обчислення додаткових витрат, доходу і прибутку, обчислення прибутку від відсотків вкладу при неперервному нарахуванні).

– **диференціальні рівняння:** мікроекономіка, макроекономіка, економічний аналіз (задачі: демографічний аналіз, аналіз ефективності реклами, аналіз зростання випуску продукції при інвестиціях, залежність національного доходу від динаміки споживання, модель ринку з прогнозованими цінами, модель зростання в умовах конкуренції).

– **ряди:** фінансова математика (задачі: на використання складних відсотків для підрахунку вартостей грошових потоків).

Важливим засобом прикладної спрямованості навчання математики майбутніх фахівців з міжнародної економіки є прикладні задачі економічного змісту. Стосовно нашого дослідження, вважаємо, можна використовувати наступне формулювання: прикладні задачі – це задачі, які виникають поза курсом вищої математики але розв'язуються з використанням математичних методів та прийомів.

Сформулюємо основні вимоги до прикладних задач, які використовуються у навчанні вищої математики [2]:

- задачі повинні мати реальний практичний зміст;
- задачі повинні відповідати навчальним програмам за формулюванням і змістом методів і фактів, які будуть використовуватися в процесі їх розв'язування;
- задачі повинні бути сформульовані доступною і зрозумілою мовою;
- числові дані в прикладних задачах повинні бути реальними, відповідати існуючим на практиці.

У деяких задачах макроекономіки ставиться питання про ефективність ведення багатогалузевого господарства. Тут кожна галузь є і виробником, і споживачем деякої продукції (як своєї, так і продукції, виробленої іншими галузями). Однак, з економічної точки зору, міжгалузевий баланс є більш ефективним у вартісному виразі. При цьому об'єднання окремих галузей у підгрупи полегшує складання балансів продукції.

Візьмемо, наприклад, тему **елементи лінійної алгебри** після того як студенти ознайомились із поняттям матриці, видами матриць, діями над матрицями, алгоритмом відшукування оберненої матриці, визначником та основними методами його обчислення їм можна запропонувати розв'язання наступних задач [3]:

- У таблиці наведено дані балансового звіту для двогалузевої моделі економіки.

Таблиця 1. Балансовий звіт для двогалузевої моделі економіки

Галузь	Споживання		Валовий продукт
	Енергетика	Машинобудування	
Енергетика	120	200	800
Машинобудування	140	180	1000

Знайти необхідний об'єм валового продукту кожної галузі, якщо кінцевий продукт $Y = \begin{pmatrix} 100 \\ 400 \end{pmatrix}$ енергетичної галузі потрібно збільшити удвічі, а машинобудування залишити на тому ж рівні.

Розв'язання. За формулою $a_{ij} = \frac{x_{ij}}{x_j}$ ($i, j = 1, 2, \dots, n$) знаходимо матрицю коефіцієнтів прямих витрат $A = \begin{pmatrix} 0,15 & 0,2 \\ 0,175 & 0,18 \end{pmatrix}$, яка є продуктивною (сума елементів кожного її стовпця менше одиниці).

Для будь-якого вектора кінцевого продукту Y вектор валового продукту X обчислюється за формулою $X = (E - A)^{-1}Y$.

Знайдемо матрицю повних витрат $S = (E - A)^{-1}$:

$$E - A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0,15 & 0,2 \\ 0,175 & 0,18 \end{pmatrix} = \begin{pmatrix} 0,85 & -0,2 \\ -0,175 & 0,82 \end{pmatrix}$$

Знайдемо обернену матрицю до матриці $E - A = \begin{pmatrix} 0,85 & -0,2 \\ -0,175 & 0,82 \end{pmatrix}$

$$|E - A| = \begin{vmatrix} 0,85 & -0,2 \\ -0,175 & 0,82 \end{vmatrix} = 0,85 \cdot 0,82 - (-0,2) \cdot (-0,175) = 0,662$$

$$S = (E - A)^{-1} = \frac{1}{0,662} \begin{pmatrix} 0,82 & 0,2 \\ 0,175 & 0,85 \end{pmatrix} \approx \begin{pmatrix} 1,24 & 0,30 \\ 1,77 & 1,28 \end{pmatrix}$$

Новий вектор кінцевого продукту $Y = \begin{pmatrix} 200 \\ 400 \end{pmatrix}$. Відповідний вектор кінцевого продукту:

$$X = SY = \begin{pmatrix} 1,24 & 0,30 \\ 1,77 & 1,28 \end{pmatrix} \begin{pmatrix} 200 \\ 400 \end{pmatrix} = \begin{pmatrix} 368 \\ 866 \end{pmatrix}.$$

Отже, валовий продукт енергетичної галузі необхідно збільшити до 368 гр.од., а машинобудівної – до 866 гр.од.

• *Лінійна модель обміну (модель міжнародної торгівлі)* дає можливість знайти співвідношення національних бюджетів країн для збалансованої торгівлі. Нехай є n країн, бюджети яких відповідно $x_1, x_2, \dots, x_n$ витрачаються на закупку товарів всередині країни або із зовні (торговий бюджет). Позначимо через a_{ij} частку бюджету j -ї країни, яка витрачається на закупку товарів у i -ї країни. Матрицю $A = (a_{ij})_{n \times n}$ називають *структурною матрицею торгівлі*. Її елементи повинні задовольняти умову

$$\sum_{i=1}^n a_{ij} = 1, j = 1, 2, \dots, n.$$

Для збалансованої торгівлі потрібно знайти вектор-стовпець $X = (x_1 \ x_2 \dots x_n)^T$, при якому

$$(A - E)X = 0,$$

де E – одинична матриця розміру $n \times n$. (3) – однорідна система лінійних рівнянь, причому її визначник $|A - E| = 0$. Тому вона має ненульовий розв'язок X .

Для структурної матриці A торгівлі трьох країн S_1, S_2, S_3 , яка має вигляд

$$A = \begin{pmatrix} 0,2 & 0,3 & 0,2 \\ 0,6 & 0,4 & 0,6 \\ 0,2 & 0,3 & 0,2 \end{pmatrix}$$

знайти співвідношення між національними доходами країн для збалансованої торгівлі.

Розв'язання. Міжнародна торгівля буде збалансованою тоді і тільки тоді, коли для кожної країни S_j , $j = 1, 2, \dots, n$, дохід $p_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n$ від зовнішньої та внутрішньої торгівлі співпадає з національним доходом x_i цієї країни, тобто коли виконується рівність $(A - E)X = 0$.

$$\text{Підставляємо дані: } \begin{pmatrix} 0,2-1 & 0,3 & 0,2 \\ 0,6 & 0,4-1 & 0,6 \\ 0,2 & 0,3 & 0,2-1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \text{ або } \begin{cases} -0,8x_1 + 0,3x_2 + 0,2x_3 = 0 \\ 0,6x_1 - 0,6x_2 + 0,6x_3 = 0 \\ 0,2x_1 + 0,3x_2 - 0,8x_3 = 0. \end{cases}$$

Розв'язуємо систему методом Гаусса:

$$\begin{pmatrix} -0,8 & 0,3 & 0,2 \\ 0,6 & -0,6 & 0,6 \\ 0,2 & 0,3 & -0,8 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -1 & 1 \\ -0,8 & 0,3 & 0,2 \\ 0,2 & 0,3 & -0,8 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -1 & 1 \\ 0 & -0,5 & 1 \\ 0 & 0,5 & -1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -1 & 1 \\ 0 & -0,5 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Перепишемо останню матрицю у вигляді системи:

$$\begin{cases} x_1 - x_2 + x_3 = 0 \\ -0,5x_2 + x_3 = 0, \end{cases}$$

$x_3 = c$, $x_2 = 2c$, $x_1 = c$, де c – довільне число.

Отже, система має розв'язок $x_1 = c$, $x_2 = 2c$, $x_3 = c$. З отриманого результату можна зробити висновок, що збалансованість торгівлі трьох країн досягається при співвідношенні національних доходів 1:2:1.

Теорія диференціальних рівнянь дає можливість розібратися у розв'язанні наступної задачі після того, як студенти ознайомились із поняттям диференціального рівняння, його розв'язку, відокремленням змінних:

У початковий момент часу $t_0=0$ кількість населення деякої країни становить P_0 . Нехай темп приросту кількості цього населення є сталим (зазначимо, що приріст може бути як додатнім, так і від'ємним) і дорівнює величині T .

Нагадавши, що темп приросту функції $y=y(t)$ обчислюється за формулою $T_y = \frac{y'}{y}$, приходимо до такої задачі Коші:

$$\begin{cases} \frac{y'}{y} = T \\ y(0) = P_0 \end{cases}$$

Розділяємо змінні і знаходимо загальний розв'язок:

$$\begin{aligned} \frac{dy}{y} &= y dt ; \\ \ln y &= T \cdot t + \ln C ; \\ y &= C \cdot e^{T \cdot t} . \end{aligned}$$

Оскільки при $t=0$ величина $y(0)=P_0$, то $P_0=Ce^{T \cdot 0}=C$ і далі $y(t)=P_0 e^{T \cdot t}$ (розв'язок задачі Коші).

Знайдена функція $y(t)=P_0 \cdot e^{T \cdot t}$ дозволяє прогнозувати кількість населення у довільний момент часу. Наприклад, при річному темпі приросту $T = -2\%$ (темпі спаду в розмірі 2%) через $t=25$ (років) кількість населення становитиме $P_0 \cdot e^{-0,02 \cdot 25} = P_0 \cdot e^{-0,5} \approx 0,607 P_0$.

Зауважимо, що ця ж функція $y(t)=P_0 \cdot e^{T \cdot t}$ описує динаміку росту цін при постійному темпі інфляції.

Розв'язування задач економічного змісту досить часто приводить до громіздких обчислень, що забирає багато аудиторного часу. Тому для швидкого отримання результату студентам пропонується використання математичного редактора Mathcad. Простота інтерфейсу середовища, широкий набір графічних, аналітичних і чисельних методів вирішення математичних завдань дозволяють ефективно використовувати його у навчальному процесі. Mathcad дозволяє виконувати як чисельні, так і аналітичні (символьні) обчислення, має зручний орієнтований інтерфейс (сукупність засобів, які забезпечують управління системою за допомогою клавіатури і миші).

ВИСНОВКИ. Базові знання з математики є основою для вивчення економічних дисциплін та важливою складовою професійної підготовки майбутніх економістів [2]. Використання прикладних задач під час вивчення курсу "Математика для економістів та економічне моделювання" для студентів міжнародників дає позитивні результати, а саме:

- сприяє розвитку творчих здібностей майбутніх фахівців;
- демонструє зв'язок теорії з практикою;
- викликає інтерес у студентів нестандартною постановкою математичного завдання;
- сприяє застосуванню математичного апарату для дослідження економічних процесів і явищ;
- допомагає побудові моделей економічних ситуацій;
- сприяє знаходженню математичних залежностей у реальних виробничих процесах.

1. Бобик О.І., Берегова Г. І. Основні принципи реалізації міжпредметних зв'язків математичних та економічних дисциплін у вузівській підготовці спеціаліста-економіста / О.І. Бобик., Г. І. Берегова // Фінансово-кредитна діяльність: проблеми теорії та практики: Зб. наук. праць. – 2015. – С. 106 – 118
2. Гусак Л.П. До питання навчання математики студентів економічних спеціальностей в умовах кредитно-модульної системи

- організації навчання / Л.П. Гусак // Сучасні інформаційні технології та інноваційні методики навчання в педагогіці фахівців: методологія, теорія, досвід, проблеми: Зб. наук. пр. Вип. 7. – Київ-Вінниця: ТОВ фірма "Планер", 2005. – С. 258 – 261.
3. Ройко Л.Л. Реалізація професійної спрямованості математичної підготовки студентів економічного профілю / Л.Л. Ройко //Збірник статей V Міжнародної науково-практичної конференції "Математика. Інформаційні технології. Освіта". № 3 – Луцьк: ПП Іванюк В. П., 2016.– С. 135 – 142
 4. Рум'янцева К.Є. Методичні рекомендації до розв'язання творчих фахових завдань з дисципліни "Математика для економістів" засобами моделювання для студентів галузі знань 0305 "Економіка і підприємництво":[метод. рек. для студ. вищ. навч. закл.] / К. Рум'янцева. – Вінниця: ВІЕ ТНЕУ, 2008. – 72 с.
 5. Самарук Н. Педагогічні умови вдосконалення математичної підготовки студентів економічного профілю / Н. Самарук // Викладач і студент: проблеми ефективної співпраці. Збірник матеріалів Всеукраїнської науково-практичної конференції. Черкаси: Видавництво ЧНУ імені Богдана Хмельницького, 2006. – С.41 – 43.
 6. Пуханова Л. С. Професійно орієнтоване навчання теорії ймовірностей і математичної статистики студентів – один із напрямків підвищення рівня якості підготовки студентів економічного профілю / Л. С. Пуханова // Навчання математики в сучасних умовах: міжнар. наук.-практ. конф., 23-25 трав. 2007р.: тези доп. – Донецьк: ДонНТУ, 2007. – С. 81 – 82.

УДК 614.878

В.Л. Филипчук, Л.В. Филипчук, О.С. Шаталов

Національний університет водного господарства та природокористування

БЕЗПЕКА ЕКСПЛУАТАЦІЇ АВТОМАТИЗОВАНИХ ОЧИСНИХ СПОРУД ДЛЯ ЗНЕШКОДЖЕННЯ СТИЧНИХ ВОД З ТОКСИЧНИМИ ОКИСНО-ВІДНОВНИМИ ДОМІШКАМИ

Филипчук В.Л., Филипчук Л.В., Шаталов О.С. Безпека експлуатації автоматизованих очисних споруд для знешкодження стічних вод з токсичними окисно-відновними домішками. В статті розглядаються процеси утворення шкідливих для обслуговуючого персоналу продуктів хімічних реакцій при очищенні стічних вод з токсичними домішками. Визначено характер впливу шкідливих речовин на працівників. Подані шляхи підвищення безпеки експлуатації локальних очисних споруд при очищенні стічних вод за умов утворення шкідливих речовин.

Ключові слова: безпека, шкідливі речовини, очистка, стічні води, токсичні домішки

Форм. 3. Табл. 1. Рис. 3. Літ. 15.

Filipchuk VL, Filipchuk LV, Shatalov AS Safety of operation of automated treatment plants for wastewater treatment with toxic redox impurities. The article deals with processes of dangerous to the service personnel chemical reactions formation in the treatment of wastewater with toxic impurities. Nature of influence of harmful substances on personnel is determined. Ways of increasing safety of local treatment facilities operation for wastewater treatment in conditions of harmful substances formation are presented.

Keywords: safety, hazardous substances, treatment, sewage, toxic impurities, automation.

Филипчук В.Л., Филипчук Л.В., Шаталов А.С. Безопасность эксплуатации автоматизированных очистных сооружений для обезвреживания сточных вод с токсичными окислительно-восстановительными примесями. В статье рассматриваются процессы образования вредных для обслуживающего персонала продуктов химических реакций при очистке сточных вод с токсичными примесями. Определен характер влияния вредных веществ на работников. Представленные пути повышения безопасности эксплуатации локальных очистных сооружений при очистке сточных вод при условиях образования вредных веществ.

Ключевые слова: безопасность, опасные вещества, очистка, сточные воды, токсичные примеси

Постановка проблеми. Під час виготовлення продукції у цехах гальваніки та друкованих плат утворюються металовмісні стічні води, що вміщують токсичні домішки, зокрема хромові та ціаністі стоки. Такі стічні води є вкрай небезпечними для обслуговуючого персоналу та довкілля. Ці стоки знешкоджуються на окремих локальних очисних спорудах, після чого скидаються у загальний потік кисло-лужних стічних вод, що утворюються на промислових підприємствах з метою спільного їх доочищення

Найбільш поширеним методом очищення хромових та ціаністих стоків є реагентний метод. Однак під час реагентного знешкодження токсичних домішок застосовується комплекс небезпечних і шкідливих реагентів та можуть виникати вторинні продукти, які при їх взаємодії вкрай негативно впливають на працівників та навколишнє середовище. Тому аналіз таких чинників та розробка заходів та засобів з попередження їх негативного впливу є достатньо актуальною задачею.

Аналіз останніх досліджень і публікацій. Найбільш небезпечними домішками у металовмісних стічних водах, що подаються на локальні очисні споруди, є ціаніди (CN^-) та шестивалентний хром ($Cr_2O_7^{2-}$), які мають окисно-відновні властивості.

Сполуки хрому(VI) є сильними окисниками. Симптоми отруєння хромом залежать від тривалості контакту з ним та шляхів надходження в організм. При попаданні на шкіру хром(VI) може викликати опіки шкіри, важке загальне отруєння. Може початися шлунково-кишкова кровотеча, некроз печінки та підшлункової залози. Найважче при отруєнні хромом уражаються органи дихання. При вдиханні хром(VI) викликає запалення, кашель, професійну бронхіальну астму, пневмосклероз і пневмоконіоз. Значно зростає ризик раку легенів [1, 2].

Знешкодження хрому(VI) проводиться шляхом його відновлення до хрому(III), який є значно менш токсичним, за допомогою реагентів-відновників. У більшості випадків в якості відновників застосовують сірковмісні реагенти, а також сполуки двовалентного заліза. Відновлення з використанням сірковмісних реагентів протікає в основному за рахунок взаємодії хрому(VI) з бісульфітом, сульфідом натрію. Відновлення проводиться у кислому середовищі при $pH < 2,5-3,0$, що потребує попереднього підкислення стічної води. Обробка стічної води включає підкислення сірчаною кислотою, уведення відновника і подальше підлогування для осадження утвореного хрому(III) у вигляді гідроксиду [3, 4].

Відновлення хрому(VI) двовалентним залізом та його сполуками може проходити в будь-якому середовищі. Найбільш розповсюдженим є відновлення у кислому або нейтральному середовищах за допомогою сірчанокислового заліза(II) з наступним уведенням лугу для осадження іонів заліза(III) і хрому(III) [5].

Реагенти, які використовуються, є небезпечними для обслуговуючого персоналу. Бісульфіт натрію може призвести до отруєння, яке супроводжується болями у шлунку, нудотою, блювотою, діареєю, сплутаністю свідомості і задухою. Сірчана кислота та сірчанокисле залізо викликають подразнення або опіки шкіри і слизової оболонки. У дихальній системі виникають хрипи, сильні болі в горлі, можливий спазм. Починаються порушення в роботі серцево-судинної системи. Для смертельного наслідку цілком достатньо, щоб всередину потрапило всього 5 мг сірчаної кислоти.

Ціаніди (синильна кислота) є надзвичайно-токсичною речовиною. Токсична дія проявляється в паралічі поглинання кисню клітинами. Цим гальмується нормальний перебіг процесу тканинного дихання. Внаслідок цього в організмі швидко формується патологічний стан, відомий під назвою тканинної або гістотоксичної гіпоксії, що проявляється задухою, важким порушенням роботи серця, судомами, паралічем і смертю [6, 7].

Для окиснення ціанідних іонів найбільш широко застосовується активний хлор (гіпохлорит натрію, діоксид хлору, хлорне вапно), менше – озон, пероксид водню тощо. Найбільш розповсюдженим окисником є гіпохлорит натрію. Окиснення ціанідів цим реагентом проводиться при рН більше 10,5. Окиснення ціанідів відбувається у дві стадії. На першій протікає процес окиснення простих або комплексних (цинкових, мідних, нікелевих) ціанідів до ціанатів, які знешкоджуються на другій стадії за рахунок гідролізу при меншому значенні рН. В результаті окиснення ціанідів утворюються азот і вуглекислота [8].

Гіпохлорит натрію є сильним окисником, який при попаданні на шкіру та слизові оболонки рота, очей, легенів руйнує їх. У легких випадках отруєння обмежується почервонінням кон'юнктиви, м'якого піднебіння і глотки. При вдиханні високих концентрацій гіпохлориту натрію розвивається так звана блискавична форма ураження. Потерпілий задихається, обличчя синіє, рухи неkoordinовані, пульс частий і потім ниткоподібний. Смерть настає швидко в результаті рефлекторної зупинки дихання та опіку легенів.

Отруєння їдким натром найчастіше настає за випадкового попадання речовини або її парів всередину організму. Їдкий натр має різко виражені властивості до руйнування тканин. Спочатку уражається слизова оболонка рота, на ній виникають виразки. Їдкий натр легко проникає крізь тканини, при цьому утворюється пухка і опікова поверхня. Також можливе ураження слизових оболонок очей [6, 7].

За ступенем впливу на організм людини серед токсичних домішок та використовуваних реагентів переважає II і III класи небезпечності – високо- та помірнонебезпечні речовини (табл. 1).

Таблиця 1. Характеристика небезпечності речовин в очисних спорудах для очищення стічних вод від ціанідів та хрому(VI)

№ з/п	Найменування речовини	Клас небезпечності	Середня концентрація у стічних водах, мг/дм ³
1.	Ціаніди	II	10-15
2.	Хром(VI)	I	15-20
3.	Гіпохлорит натрію	II	40-60
4.	Бісульфіт натрію	III	90-110
5.	Їдкий натрій	II	60-80
6.	Сірчана кислота	II	10-20
7.	Сірчанокисле залізо(II)	III	75-110
8.	Гідроксиди заліза(III) та хрому(III)	III	45-70

Під час очищення ціаністих та хромових стічних вод застосовують в першу чергу засоби індивідуального захисту. Всі працівники, повинні бути забезпечені спецодягом, спецвзуттям та індивідуальними засобами захист. Одяг та взуття при виконанні робіт, що вимагають контакту з реагентами, видається спеціальний, з підвищеним ступенем захисту. Обов'язковими є окуляри, респіратори і маска з оргскла. Для надання екстреної допомоги працівнику при отруєнні,

попаданні стічних вод або реагентів на шкіру та слизові оболонки на робочих місцях повинні бути встановлені висхідні фонтанчики, раковини самопомоги, аварійні душі [11-13].

Значна увага приділяється очищенню повітря у робочій зоні і приміщеннях. Улаштовується загальна припливно-витяжна вентиляція, а також за необхідності місцеве відведення забрудненого повітря із робочої зони за допомогою витяжного обладнання, зокрема у вигляді бортових або парасольних відсмоктувачів [14].

В той же час, під час очищення токсичних стічних вод, змішування знешкоджених стоків або недотримання параметрів технологічного процесу можуть утворюватися вторинні токсичні продукти, які негативно впливають на працівників як очисних споруд, так і інших виробництв і потребують додаткових заходів безпеки.

Невирішені частини проблеми. Являючись одним із найбільших споживачів води, гальванічне виробництво має найбільш токсичні і шкідливі стічні води. Українськими та закордонними вченими (В. М. Рогов, А. К. Запольський, Г. С. Пантелют, В. Є. Терновцев, Г. М. Кочетов, М. Д. Гомеля, В. Є. Генкін, Ф. П. Волоцков, Л. В. Мілованов, G.C. Cushni, J. W. Patterson, J. V. Rouse, J. Maly та ін.) постійно висвітлюються різноманітні проблеми пов'язані з очищенням стічних вод та безперервно удосконалюються його технології і установки. Однак питання, присвячені безпечній експлуатації установок для очищення стічних вод в залежності від впливу на обслуговуючий персонал токсичних речовин, які можуть виникнути внаслідок комплексного використання різних реагентів, в літературі практично не розглядаються.

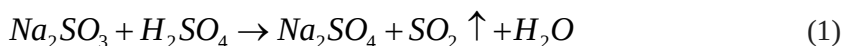
Метою дослідження є оцінка можливості утворення побічних вторинних продуктів при очищенні ціаністих та хромовмісних стічних вод від хрому(VI) та ціанідів і розробці додаткових технічно-організаційних заходів з безпечної експлуатації локальних очисних споруд промислових підприємств.

Методика досліджень. Досягнення мети реалізовувалось на підставі фізичного моделювання процесів очищення хромових та ціаністих стічних вод, що вміщують домішки з окисно-відновними властивостями, з використанням відомих рівнянь окисно-відновних процесів, чисельних та аналітичних методів визначення параметрів технологічних процесів, дослідних даних, одержаних в лабораторних та виробничих умовах.

При проведенні досліджень для визначення параметрів активної реакції рН та окисно-відновного потенціалу E_h , концентрацій важких металів та інших домішок застосовані стандартні методики проведення хімічних аналізів: колориметричний та потенціометричний. Лабораторні і дослідно-виробничі дослідження проводились з використанням сертифікованих методик фізико-хімічних аналізів та експериментальної лабораторної бази НУВГП та закордонних підприємств.

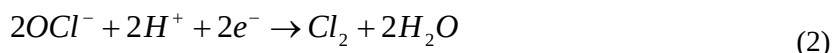
Інформаційною основою для оцінки і розробки необхідних засобів та заходів з безпечної експлуатації споруд для знешкодження стічних вод, що вміщують ціаніди та хром(VI), слугували державні будівельні норми, нормативно-правові акти з охорони праці та практичний досвід, отриманий авторами на промислових підприємствах різних країн, зокрема «US Steel», «Lukas autobrzdy», «Volkswagen» тощо, під час експлуатації установок для знешкодження токсичних домішок у стічних водах різноманітного складу.

Основні результати дослідження. Під час відновлення хрому(VI) за допомогою сірчаноокислого заліза(II) у воді утворюються іони та гідроксиди хрому(III) та заліза(III), які мають III та IV класи небезпеки, що надалі вилучаються під час фільтрування після підлужування обробленої стічної води до рН 9-10. При застосуванні сірковмісних реагентів у кислому середовищі із стічної води можливе виділення значних кількостей діоксиду сірки (сірчастого газу) SO_2 , особливо при зниженні рН середовища нижче 3-4 [8].



Діоксид сірки є токсичною речовиною. Симптоми при отруєнні – нежить, кашель, захриплість, першіння у горлі. При вдиханні сірчастого газу більш високої концентрації – задуха, розлад мови, блювання, можливий гострий набряк легенів [6, 7].

Під час знешкодження ціаністих стоків гіпохлоритом натрію у стічну воду необхідно подавати значний надлишок цього реагенту, що забезпечує повне окиснення високотоксичних ціанідів. При зниженні рН в процесі хімічної реакції або попаданні у змішувач-реактор кислих стічних вод при рН менше 6,7 можливе утворення газоподібного хлору, який є сильним окисником, що особливо небезпечно впливає на слизові оболонки та має задушливу дію.



Під час зниження рН також можливе протікання реакції прямого хлорування ціанідів з утворенням надзвичайно токсичного хлорціану CNCl [8]:



Хлорціан – безбарвна рідина з гострим запахом. Надходить в організм через органи дихання, шлунково-кишковий тракт, дуже швидко всмоктується через слизові оболонки. Уражається центральна нервова система, страждають дихальні та судинорухові центри. При хронічному отруєнні знижується діяльність щитовидної залози. При високих концентраціях настають судоми, втрата свідомості та миттєва смерть внаслідок паралічу дихання або паралічу серця [13]. Утворені газоподібний хлор відноситься до II класу, сірчистий газ – до III класу, хлорціан – до II класу небезпечності.

При змішуванні знешкоджених стічних вод із загальним потоком кисло-лужних стоків можливо утворення ряду небезпечних речовин. Оскільки знешкоджені ціаністі стічні води мають у своєму складі надлишок гіпохлориту натрію, то при змішуванні із знешкодженими хромовими стічними водами, що вміщують тривалентний хром, можливе протікання зворотної реакції окиснення хрому(III) у токсичний хром(VI), особливо в лужному середовищі, яке створюється для осадження іонів важких металів. Це підтверджується розрахунками із використанням значень окисно-відновного потенціалу E_h , який можна використовувати для оцінки протікання окисно-відновних реакцій знешкодження токсичних домішок.

Розрахунки показують, що окиснення хрому(III) у стічній воді з $\text{pH}=8$ можливе при величині $E_h > +0,6\text{В}$, з $\text{pH}=9$ – при $E_h > +0,5\text{В}$, з $\text{pH}=10$ – при $E_h > +0,4\text{В}$. Експериментальні результати наших досліджень підтверджують, що при змішуванні знешкоджених хромвмісних і ціаністих стічних вод при рН середовища у межах 8,5-9,5 і E_h у межах $+(0,5-0,7)\text{В}$ відбувається часткове зворотне окиснення хрому(III), в результаті чого концентрація хрому(VI) збільшується до 0,8-2,2 мг/дм³ при їх допустимих концентраціях у воді, менше 0,1 мг/дм³.

Для запобігання окисненню хрому(III) при змішуванні знешкоджених хромвмісних стічних вод та інших стоків з високими значеннями E_h потрібно проводити їх змішування перед відновленням хромвмісних стічних вод. Це дозволить нейтралізувати окисні властивості інших стоків за рахунок відновників, що використовуються для відновлення хрому(VI). При неможливості такого змішування потрібна спеціальна попередня нейтралізація окисних властивостей стоків для зниження E_h середовища до $+(0,2-0,3)\text{В}$ за допомогою введення відновників і потенціометричного контролю за величиною E_h .

Існуючі засоби індивідуального захисту забезпечують безпеку працівників при виділенні інших токсичних речовин. Технічні та інші вимоги до трубопроводів, арматури, насосів, насосів-дозаторів, приміщень тощо повинні відповідати прийнятим нормативним документам [9-14]. Додаткові заходи безпеки під час експлуатації локальних очисних споруд знешкодження хромових та ціаністих стоків відносяться до технічно-організаційних.

Важливим заходом з мінімізації виділення токсичних продуктів хімічних реакцій при очищенні стічних вод є необхідність підтримання у заданих межах основних параметрів знешкодження, зокрема величин рН та окисно-відновного потенціалу E_h водного середовища. Це можна досягти при застосуванні у технологічній схемі очищення стічних вод непроточних змішувачів-реакторів, яких повинно бути не менше двох, щз працюють поперемінно. В першому змішувачі-реакторі проводиться процес знешкодження стічної води, а у другому – її накопичення. Цикл обробки стічної води у змішувачі-реакторі включає: накопичування стічних вод, дозування реагентів, їх змішування, протікання хімічних процесів знешкодження та відкачування знешкодженої стічної води. Оскільки у змішувачі-реактори повинні підтримуватись значення рН та окисно-відновного потенціалу E_h у вузькому діапазоні і дозуватись точні кількості реагентів, то ці ємкості повинні бути обладнанні датчиками вимірювання рН та окисно-відновного потенціалу E_h стічної води.

Змішувачі-реактори закриваються кришками, обладнуються витяжною вентиляцією, а також вимірювачами рівня розчинів, пристроями для запобігання перепоовненню ємкостей. Найефективнішим з точки зору відсутності контакту обслуговуючого персоналу з токсичними реагентами є застосування мобільних баків з готовими розчинами потрібної концентрації заводського приготування, які починають широко використовуватись на закордонних підприємствах (рис. 1).

Для попередження отруєння газами, що виділяються, необхідно провести герметизацію емкостей, арматури, дозуючих пристроїв. Слід влаштувати додаткову автоматичну сигналізацію для контролю небезпечних концентрацій токсичних речовин у повітряному середовищі робочих приміщень, зокрема хлору та хлорціану.

Головна увага повинна приділятися повній автоматизації процесу знешкодження стічних вод. Для цього рекомендується використання ступінчастого дозування реагентів для зміни рН та Eh у потрібному напрямку [15]. Якщо необхідне окислення домішок застосовують окисні реагенти з поступовим підвищенням Eh, якщо відновлення – відновлювальні реагенти з поступовим зниженням Eh до необхідного значення. В залежності від хімічних властивостей окисника або відновника для регулювання рН застосовують луг або кислоту.



Рис. 1. Змішувач-реактор непроточного типу та реагентні баки заводського виготовлення з насосами-дозаторами реагентів для очищення стічних вод від шестивалентного хрому підприємства «Lukas autobrzdy» (м. Яблонець над Нісою, Чехія) (власна розробка)

Система автоматизації процесу реагентного знешкодження домішок у стічних водах включає вимірювання параметрів рН та Eh, формування керуючого сигналу для дозування у воду реагентів виконавчим механізмом, яке здійснюють на базі адаптивної системи автоматизованого регулювання з використанням регулятора на основі нечіткої логіки [16]. Далі проводять, ступінчасте дозування реагентів, на кожному ступені якого на базі промислового контролера програмно задають необхідні параметри дозування спочатку лугу (кислоти) для зміни рН до величини, оптимальної для реакції окиснення (відновлення), а потім окисника (відновника) для підвищення (зниження) величин Eh до значення, необхідного для протікання реакції окиснення (відновлення). Потім проводять перемішування води з реагентами для руйнування токсичних домішок з супутнім зниженням (збільшенням) рН та Eh. При цьому на кожному ступені дозування окисника (відновника) зміну Eh проводять до проміжних значень цієї величини з поступовим наближенням Eh до кінцевого значення, яке визначається окисно-відновним потенціалом окисника. Кожний ступінь наступного дозування лугу та окисника починають після супутнього зниження величини рН на попередньому ступені до її граничного значення, яке визначається областю хімічної стабільності окисника, що виключає утворення вторинних токсичних продуктів реакції.

Зокрема, при знешкодженні ціанідів спочатку у стічну воду дозують луг для підвищення величини рН до 10,5-10,8, після чого вводять гіпохлорит натрію для збільшення величини Eh до значень на 200-300 мВ більше, ніж вихідне значення. Після перемішування і стабілізації параметрів рН стічної води при окисненні знижується. У воду знову дозують луг для підвищення рН до 10,5-10,8, в результаті чого Eh знижується, і далі додають окисник для підвищення Eh до 400-450 мВ з подальшим додаванням лугу після стабілізації параметрів рН та Eh. Далі ступінчасте додавання реагентів проводять до підвищення величини Eh до 600-700 мВ і потрібного кінцевого значення 800-900 мВ із стабілізацію контрольних параметрів. Кількість ступенів додавання

реагентів приймають 4-5. Якщо Eh води на протязі 10-15 хвилин не знижується нижче +800-850 мВ, реакція окислення ціанідів вважається закінченою. Для гарантії окислення ціанідів після кінцевої стабілізації Eh проводиться хімічний або потенціометричний контроль наявності залишкового активного хлору, концентрація якого повинна бути не менше 2-3 мг/л. Далі оброблена стічна вода направляється у відстійник і фільтр для вилучення утвореної зависі.

Рекомендована технологічна схема очистки ціаністих стічних з автоматизованою системою дозування реагентів наведена на рис. 2.

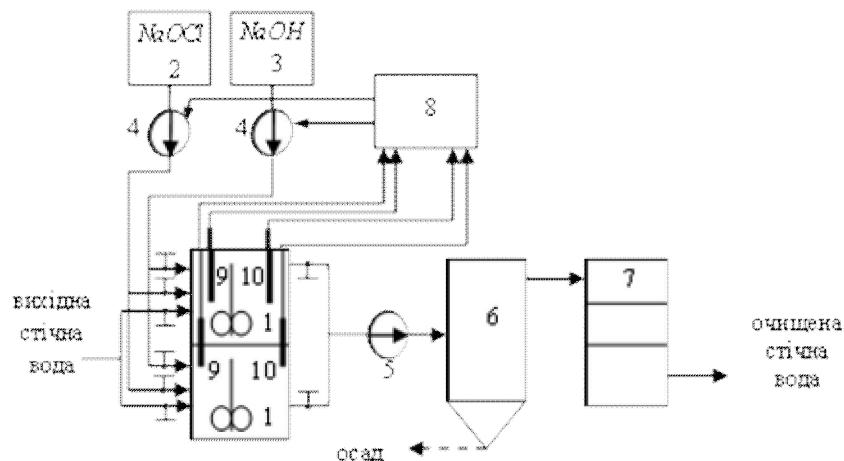


Рис. 2. Принципова технологічна схема знешкодження ціаністих стічних вод з регулюванням рН та Eh:

1 – механічні непроточні реактори-змішувачі; 2 – вузол приготування розчину NaOCl ; 3 – вузол приготування розчину NaOH ; 4 – насоси-дозатори реагентів; 5 – насос; 6 – відстійник; 7 – фільтр; 8 – промисловий контролер; 9 – датчики рН; 10 – датчики Eh

Процес ступінчастого автоматичного керування процесом очищення проводиться за допомогою промислового логічного контролера (ПЛК) фірми Vipa, для якого розроблялось програмне забезпечення. ПЛК в автоматичному режимі обробляє отримані з датчиків рН і Eh дані та подає керуючий сигнал на дозування необхідної кількості реагентів до заданого значення рН і відповідних проміжних значень Eh в залежності від кількості вибраних циклів обробки води у змішувачі-реакторі. Контролер використовує ПІД-закон регулювання. Це дозволяє більш плавно та швидко регулювати рН та Eh, суттєво зменшити перерегулювання і за рахунок цього оптимізувати витрату реагенту в залежності від вихідних показників якості стічної води. Крім того, контролер керує роботою механічних мішалок, насосів для подачі вихідної та перекачки обробленої стічної води, вентилів, що переключають подачу стічної води та розчинів реагентів у змішувачі-реактори і відстійник та випускають осад.

Застосування розробленої системи автоматизованого регулювання рН та Eh в технологічній схемі знешкодження стічних вод з токсичними окисно-відновними домішками дозволяє мінімізувати кількість використаних реагентів, повністю автоматизувати процес очищення і забезпечити надійність окислення незалежно від складу стічних вод та коливань концентрацій домішок. Це надає можливість виключити прямий контакт обслуговуючого персоналу з токсичними реагентами та домішками, мінімізувати можливість утворення побічних вторинних продуктів хімічних реакцій і забезпечити необхідну безпеку експлуатації очисних споруд.

Висновки. Аналізуючи вище викладене, можна констатувати, що при експлуатації локальних очисних споруд для знешкодження ціаністих та хромвмісних стічних вод можливе утворення небезпечних для обслуговуючого персоналу продуктів хімічних реакцій, таких як хлорціан, газоподібний хлор, діоксид сірки, вторинний хром(VI), наявність яких потребує додаткових заходів з безпеки праці у водоочисній станції.

Для підвищення безпеки експлуатації очисних споруд, унеможливлення контакту обслуговуючого персоналу з токсичними продуктами необхідно в автоматичному режимі за допомогою промислового логічного контролера та розробленого програмного забезпечення проводити ступінчасте дозування хімічних реагентів для регулювання активної реакції рН та

окисно-відновного потенціалу Eh стічних вод із застосуванням непроточних механічних змішувачів-реакторів.

Подальшим напрямком досліджень є оцінка можливості утворення побічних токсичних продуктів під час очищення багатокomпонентних металовмісних стічних вод від виробництва друкованих плат і розробка відповідних заходів з безпеки праці при експлуатації очисних споруд.

1. Безак-Мазур, Е. Транскордонні проблеми токсикології довкілля [Текст] / Е. Безак-Мазур, Т. Шендрік. - Донецьк : ГП "Інформ.-аналитический центр "Донбассинформ" 2008, - 300с.
2. Коломоець М.Ю. Професійні хвороби / М.Ю. Коломоець, О.С. Хухліна. - К.: Здоров'я, 2004. - 628 с.
3. Fenglian F. Removal of heavy metal ions from wastewaters. / F. Fenglian, Q. Wang. // Journal of Environmental Management, Volume 92, Issue 3, 2011, P. 407-418
4. Maly J. Cistení průmyslových odpadních vod / J. Maly, P. Hlavinec. - Brno: NOEL 2000, 1996. - 255 s.
5. Kim C. Cr6+ reduction by hydrogen sulfide in aqueous media: stoichiometry and kinetics / C. Kim, Q. H. Zhou, B. L. Deng // Environ. Sci. Technol., 2001.- 35. - P. 2219-2225.
6. Вредные химические вещества / Под общей ред. В.А. Филова. - Л.: Химия, 1990. - 734 с.
7. Общая токсикология / под ред. Б. А. Курляндского. - М.: Медицина, 2002. - 614 с.
8. Проскуряков В.А. Очистка сточных вод в химической промышленности / В. А. Проскуряков, Л. И. Шмидт. - Л. : Химия, 1977. - 464 с.
9. НПАОП 73.1-1.11-12 Правила охорони праці під час роботи в хімічних лабораторіях / наказ Міністерства надзвичайних ситуацій України від 11.09.2012 № 1192.
10. НПАОП 28.51-1.16-85 Правила з техніки безпеки при травленні металів і нанесенні на них гальванічних і хімічних покриттів. - М.: Міноборонпром СРСР, 1985.
11. НПАОП 0.00-8.11-12 Вимоги до роботодавців щодо захисту працівників від шкідливого впливу хімічних речовин / наказ Міністерства надзвичайних ситуацій України від 22.03.2012, № 627.
12. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування. - Київ, Мінрегіон України, 2013. - 141 с.
13. Филипчук Л. В. Автоматизація процесу регулювання рН та Eh при очищенні металовмісних стічних вод в оборотних системах водопостачання підприємств / Филипчук Л. В. // Вісник Інженерної академії України. - Київ, 2013. - Вип. 1, 2013. - С. 204-208.
14. Клепач М. І. Застосування нечіткої логіки для автоматизації процесу регулювання рН стічних вод в гідравлічному змішувачі-реакторі / М. І. Клепач, Л. В. Филипчук // Інтегровані інтелектуальні робототехнічні комплекси ПРТК-2012: 5 міжнародна науково-практична конференція, 15-16 травня 2012 р. - Київ : НАУ. - С. 274-275.

УДК 004:338

Черніков М.М., Куваєва В.І., Болтъонков В.О., к.т.н.
Одеський національний політехнічний університет

СИСТЕМА РОЗПОДІЛЕНОЇ КОЛЕКТИВНОЇ ЕКСПЕРТИЗИ З ЗАХИСТОМ ВІД ФАЛЬСИФІКАЦІЙ

Черніков М.М., Куваєва В.І., Болтъонков В.О. Система розподіленої колективної експертизи з захистом від фальсифікацій. Розроблено систему колективного розподіленого експертного оцінювання, захищену від фальсифікацій. Побудова системи на основі блокчейн-технологій гарантує неможливість внесення цілеспрямованих змін як в індивідуальні експертні думки, так і в результати експертизи в цілому. Система ефективна в умовах участі в експертизі необ'єктивних експертів або корумпованих організаторів експертизи. Система практично реалізована на платформі Ethereum.

Ключові слова: розподілена колективна експертиза, захист від фальсифікацій, блокчейн-технології, Ethereum.

Черников М.Н., Куаева В.И., Болтенков В.А. Система распределения коллективной экспертизы с защитой от фальсификаций. Разработана система коллективного распределенного экспертного оценивания, защищенная от фальсификаций. Построение системы на основе блокчейн-технологий гарантирует невозможность внесения целенаправленных изменений как в индивидуальные экспертные мнения, так и в результаты экспертизы в целом. Система эффективна в условиях участия в экспертизе необъективных экспертов или коррумпированных организаторов экспертизы. Система практически реализована на платформе Ethereum.

Ключевые слова: распределенная коллективная экспертиза, защита от фальсификаций, блокчейн-технологии, Ethereum.

Chernikov M.M., Kuvaieva V.I., Boltanov V.A. System of distributed collective expertise with protection from falsifications. A system for collective distributed expert estimation, protected from falsifications, has been developed. The construction of a system based on blockchain technologies guarantees the impossibility of making purposeful changes both in individual expert opinions and in the results of the expertise as a whole. The system is effective in terms of participation in the examination of biased experts or corrupt expertise organizers. The system is practically implemented on the Ethereum platform.

Keywords: distributed collective expertise, protection against falsifications, blockchain technology, Ethereum.

Постановка проблеми. У прийнятті складних рішень важливу роль відіграють процедури колективного експертного оцінювання – експертизи. З початку ХХІ століття поширилася розподілена мережева експертиза, що використовує сучасні інформаційні технології та мережі зв'язку і передачі даних. Перевагами такої експертизи є оперативність, точність, репрезентативність думок експертів, можливість безпосередніх комунікацій між експертами [1,2,3].

У той же час широке поширення дистанційних експертних систем створює більше можливостей для фальсифікації результатів експертного оцінювання. Зокрема, участь в процесі експертизи корумпованих експертів або модераторів експертизи дозволяє спотворити результати колективної оцінки в інтересах певних кіл, пов'язаних з організацією оцінювання варіантів рішення. С [4,5]. Тому розробка розподілених експертних систем, захищених від фальсифікацій є актуальною практичною проблемою.

Аналіз останніх досліджень та публікацій. Більшість публікацій, присвячених захисту систем колективної експертизи від фальсифікацій, спрямовані на розробку алгоритмів, що дозволяють виявити експертів, оцінки яких істотно відрізняються від середньої експертної думки шляхом застосування спеціальних алгоритмів обробки експертних оцінок [6,7]. Іншим варіантом є застосування для експертизи процедури Коупленда, фальсифікація в якій є NP-повною задачею і вимагає великих витрат обчислювального часу [8,9]. Незважаючи на складність запропонованих алгоритмів зазначені методи все одно не гарантують повного захисту системи від фальсифікацій.

Мета. Метою роботи є побудова системи розподіленої колективної експертизи, що максимально захищена від фальсифікацій, з застосуванням блокчейн-технологій.

Виклад основного матеріалу та обґрунтування результатів дослідження. Для захисту від фальсифікацій створено систему, колективної експертизи на основі технології блокчейну [10].

Блокчейн, тобто ланцюжок блоків транзакцій – є розподілена база даних, яка підтримує перелік записів, так званих блоків, що постійно зростає. База захищена від підробки та переробки. Кожен блок містить часову мітку та посилання на попередній блок хеш дерева.

Блок транзакцій – спеціальна структура для запису групи транзакцій. Щоб транзакція вважалася достовірною («підтвердженою»), її формат і підписи повинні бути перевірені, лише після цього групу транзакцій записати в спеціальну структуру – блок. Інформацію в блоках можна швидко перевірити. Кожен блок завжди містить інформацію про попередній блок. Усі блоки

можна вибудувати в один ланцюжок, що містить інформацію про всі вчинені коли-небудь операції. Перший блок в ланцюжку – первинний блок – розглядається як окремий випадок, оскільки в нього відсутній материнський блок. Якщо використовувати систему блокчейн як місце зберігання даних, можливо забезпечити гарантований захист від зміни даних після запису у систему. Також, якщо використати якусь загальну систему та зробити колективну експертизу відкритою (але сховати справжні імена експертів за їх унікальними адресами), можливо надати доступ контролюючим особам для перевірки даних через стандартні методи блокчейну, код яких відкритий та перевірений. Для вирішення задачі обрано блокчейн-систему Ethereum [12-13]. Ethereum (укр. Етеріум) – платформа для створення практично будь-яких децентралізованих онлайн-сервісів на базі блокчейна, що працюють на базі розумних контрактів. Ethereum реалізована як єдина децентралізована віртуальна машина. Оскільки Ethereum сильно спрощує і здешевлює впровадження блокчейна, його широко використовують сьогодні багато великих корпорацій для самих різних цілей. Для того, щоб транзакція потрапила до блоку, необхідно провести процес майнінгу транзакції. Майнінг – це діяльність з підтримки розподіленої платформи і створення нових блоків з можливістю отримати винагороду в формі емітованої валюти і комісійних зборів у різних криптовалютах. Вироблені обчислення потрібні для забезпечення захисту від повторного використання одних і тих же одиниць валюти, а зв'язок майнінгу з емісією стимулює людей витратити свої обчислювальні потужності і підтримувати роботу мереж. Але звичайний майнінг підходить для дуже великої системи і зовсім не підходить до локального блокчейну, тому що не потрібно видобувати нову валюту та використовувати великі потужності для підтримки усієї системи в цілому. Існує декілька різних способів додавання транзакцій до блоку. Звичайний майнінг має загальну назву PoW (Proof of Work)[12,13].

Доказ виконання роботи (PoW) – це принцип захисту систем від зловживання послугами. Він заснований на необхідності виконання стороною, що робить запит (клієнтом) деякої досить складної тривалої роботи (PoW-завдання, одностороння функція), результат якої легко і швидко перевіряється стороною, що обробляє запит (сервером). Головна особливість цих схем полягає в асиметрії витрат часу – досить велика тривалість для ініціатора запиту і висока швидкість для відповіді. Подібні схеми також відомі як Client Puzzle (функція клієнтської головоломки), Computational Puzzle (обчислювальна головоломка), або CPU pricing function [12].

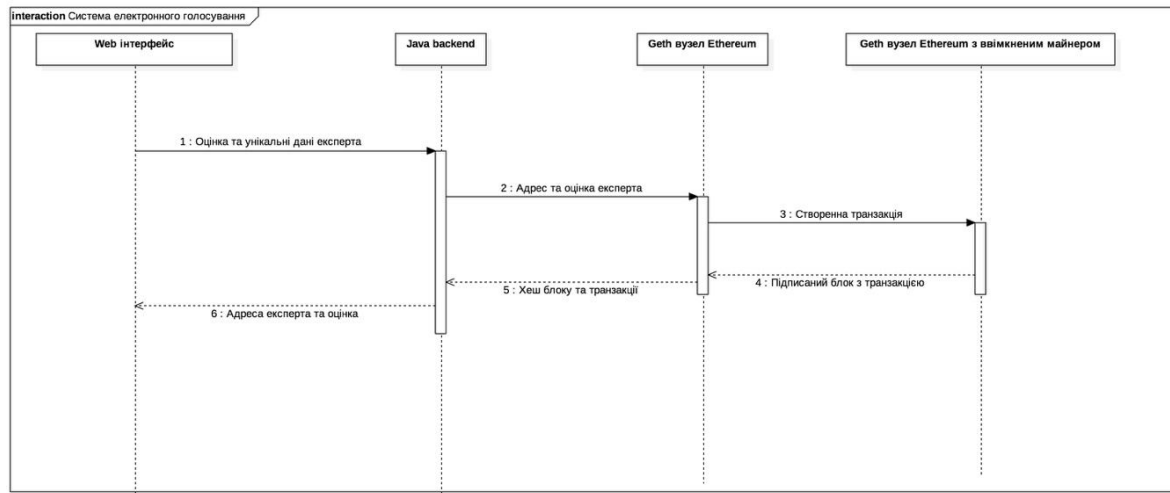
Метод підтвердження частки (PoS – Proof of Stake) – метод захисту в Ethereum, заснований на необхідності доказу зберігання певної кількості коштів на рахунку. При використанні цього методу алгоритм криптовалюти з більшою ймовірністю вибере для підтвердження чергового блоку в ланцюжку обліковий запис з великою кількістю коштів на рахунку. Метод використовують як альтернативу методу PoW (доказ виконання роботи), в якому більшу ймовірність підтвердження блоку має обліковий запис з великими обчислювальними потужностями.

Метод авторизованих користувачів (PoA – Proof of Activity) – метод захисту в розподілених блокчейн системах, де можливість додавання транзакцій до блоків мають певні авторизовані вузли. Для локальної системи блокчейну найкраще підходить метод PoA, тому що усі вірні транзакції будуть додані до блоків, список майнерів додається до першого блоку (genesis) та не може бути змінений без доступу до активних майнінг вузлів.

Для створення алгоритму колективної дистанційної експертизи та зберігання даних створюється смарт-контракт. Смарт-контракт – це комп'ютерний протокол, який спрощує, верифікує, або забезпечує дотримання переговорів, або виконання договору, перевіряє непотрібні пункти договору. Смарт-контракти, як правило, мають інтерфейс користувача і часто слідує логіці договірних положень. Прихильники розумних контрактів стверджують, що таким чином багато видів договірних положень може бути здійснено частково або повністю, самостійно або вдвох. Смарт-контракти спрямовані на забезпечення безпеки, яка перевершує традиційне договірне право, а також на зменшення операційних витрат. Смарт контракт додається до системи один єдиний раз та його код не може бути змінений після цього. Він зберігається на окремій адресі та усі транзакції, в яких виконуються методи смарт-контракту спрямовані до цієї адреси.

Послідовність всього процесу отримання оцінки експерта зображена на рис.1.

Спочатку дані надходять від користувача, що заповнив їх у веб-інтерфейсі, до серверу (Java backend). Дані перевіряються на правильність та можливість експерта поставити оцінку. Якщо усі дані вірні, сервер відправляє їх на зберігання до блокчейну, використовуючи методи смарт-контракту. Сам



До цього genesis-блоку було додано інформацію про декілька адрес, що мають можливість додавати транзакції до блоків, та підписувати їх. Також цим адресам було додано початкову кількість умовних монет (одиниць ресурсу) Ether. Наприклад, адреса 08a58f09194e403d02a1928a7bf78646cfc260b0 додана до поля “extra data” для підключення в список авторизованих адресів та в список “alloc” для надання початкових монет. Поле clique вказує, що створений блокчейн працює на основі методу PoA та буде збирати блоки кожну секунду (за це відповідає параметр period). Усі інші параметри є автогенерованими для того, щоб перший блок не відрізнявся по структурі від наступних. Далі для запуску вузлів geth на основі цього genesis блоку необхідно додати цей блок до кожного з вузлів за допомогою команди geth init. Виклики методів смарт-контракту також є транзакціями, які повинні потрапляти до блоку.

```

address[] public votersAddresses;
address[] public votedAddresses;
mapping(address => uint) public voterMarks;
function Voting(address[] _voters) public {
    for (uint i = 0; i < _voters.length; i++) {
        votersAddresses.push(_voters[i]);
    }
}

```

```
function vote(uint mark) public ableToVote {
    voterMarks[msg.sender] = mark;
    votedAddresses.push(msg.sender);
}
```

```
modifier ableToVote {
    bool flag = false;
    // Check if address is in voters list;
    for (uint i = 0; i < votersAddresses.length; i++) {
        if (msg.sender == votersAddresses[i]) flag = true;
    }
    if (flag == false) revert();
    // Check if not voted yet;
    if (votedAddresses.length != 0) {
        for (uint k = 0; k < votedAddresses.length; k++) {
```

```

    if (msg.sender == votedAddresses[k]) revert();
  }
}
}

```

Для з'єднання серверу з вузлом geth служить RPC інтерфейс Ethereum, що має назву web3 та строгий список методів. На сервері використовується бібліотека web3j, яка є імплементацією web3 на мові програмування Java. Сервер та geth спілкуються через текстові повідомлення з використанням протоколу HTTP. Основними повідомленнями серверу, які відправляються до вузла, є створення контракту, створення користувачів з адресами, та виклики методів смарт-контракту (Наприклад, метод vote). Для прийняття рішення можливо використати будь-який з відомих методів, або навіть декілька, оскільки у контракті зберігається масив усіх оцінок. Метод обробки результатів колективної експертизи можна додати до коду контракту, або до коду серверної сторони проекту. Це може бути метод медіанних оцінок [14], або один з методів соціального вибору для отримання консенсусного колективного рішення [15]. Щоб перевірити вірність винесеного рішення можна підключитись до будь-якого вузла блокчейну в системі через стандартний клієнт Ethereum, який має назву Mist, отримати масив оцінок експертів та підрахувати результат незалежним методом, або викликати метод підрахування, якщо його реалізовано на стороні коду контракту. Результати вірного функціонування розробленої системи колективної дистанційної експертизи перевірено та підтверджено на низці тестових та практичних прикладів експертного оцінювання.

Висновки. Фальсифікація результатів колективної дистанційної експертизи, що можлива при наявності в складі експертної групи необ'єктивних або корумпованих експертів або модераторів експертизи, є серйозною практичною проблемою. Для вирішення цієї проблеми запропонована побудова системи експертизи на основі блокчейн-технології. В якості блокчейн-інструменту використано платформу Ethereum. Практична реалізація виконана з використанням мови програмування Solidity. Колективна експертиза створена як розумний контракт. Практична реалізація системи підтвердила її працездатність та захищеність від можливих фальсифікацій.

1. Gubanov D., Korgin N., Novikov D., Raikov A. E-Expertise: Modern Collective Intelligence. — Springer International Publishing, Switzerland, 2014. — 112 p.
2. Масыч М.А., Целых А.А. Методы и технологии проведения сетевой экспертизы инновационных проектов // Современные проблемы науки и образования. — 2013. — № 6. [Електронний ресурс] / Режим доступу: <http://www.science-education.ru/ru/article/view?id=11642> — Назва з екрану.
3. Newman M.E.J. The Structure of Scientific Collaboration Networks // Proc. Natl. Acad. Sci. USA. — 2001. — Vol. 98. — Pp. 404—409.
4. Zhu Y., Truszczynski M. Manipulation and Bribery When Aggregating Ranked Preferences. // In: Walsh T. Algorithmic Decision Theory. Lecture Notes in Computer Science, Vol/ 9346. — Springer: Cham, 2015. — Pp. 86-102.
5. Beliaikov G., James S., Smith L., Wilkin T. Biased experts and similarity based weights in preferences aggregation. // In: EUSFLAT 2015: Proceedings of the 16th World Congress of the International-Fuzzy-Systems-Association (IFSA). — Amsterdam : Atlantis Press. — 2015. — Pp. 363-370.
6. Rezvani M., Allahbakhsh M., Vigentini L., Ignjatovic A., Jha S. An Iterative Algorithm for Reputation Aggregation in Multi-dimensional and Multinomial Rating Systems. // In: Federrath H., Gollmann D. (Eds.) ICT Systems Security and Privacy Protection. SEC 2015. IFIP Advances in Information and Communication Technology, vol 455. Springer, Cham, 2015. — Pp. 189-203.
7. Hoffman K., Zage D., Nita-Rotaru C. A Survey of Attack and Defense Techniques for Reputation Systems // ACM Comput. Surv. — 2009. — №42(1). — Pp.1-31.
8. Davies J., Katsirelos G., Narodytska N., Toby Walsh T., Xia L. Complexity of Algorithms for the Manipulation of Borda, Nanson's and Baldwin's Voting Rules // Artificial Intelligence. — 2014. — Vol. 217. — P. 20-42.
9. Faliszewski P., Hemaspaandra T., Schnoor H. Manipulation of Copeland Elections // In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1. — Toronto, 2010. — Pp.367-374.
10. Bambara J. J., Allen P. R. Blockchain. A practical guide to developing business, law, and technology solutions. — McGraw-Hill Education, 2018 — 302 p.
11. Prusty N. Building Blockchain Projects. Develop real-time practical DApps using Ethereum and JavaScript. — Birmingham — Mumbai: Packt Publ., 2017. — 245 p.
12. Bashir I. Mastering Blockchain. — Birmingham: Packt Publ., 2017 — 540 p.
13. Болтенков В.А., Куваева В.И., Позняк А.В. Анализ медианных методов консенсусного агрегирования ранговых предпочтений // Информатика та математичні методи в моделюванні. — 2017. — Том 7, №4. — С. 307-317.
14. Болтенков В.А., Куваева В.И., Червоненко П.П. Применение методов социального выбора в задачах агрегирования оценок в ранговых шкалах // Системные технологии. — 2018. — Вып. 2 (115). — С.93 — 102.

CONTENTS

INFORMATICS AND COMPUTER SCIENCE

PhD NATALIYA BAHNIUK, PAVLO MYKHAILUTSA, ANDRII KHRISTINETS Web service authorization for local network users.	5
BORTNYK K.YA., LOMINSKAYA G.YU. Technologies for analyzing the effects of cyber attacks.	10
BORTNYK K.Ya., OLSHEVSKY O.V., PASHCHUK V.Y. Internet of things and how it will change our life in the future.	14
GNENNY A.P, GORDIENKO Y. Actual mechanisms of personality identification on the Internet of things.	19
HRYNYUK S.V. POLISHCHUK M.M, MISKEVICH O.I., KHARKOVETS R.V. Software for encrypting and decrypting information by cryptographic methods with Visual Studio tools.	26
HUBAL H.M. Using some commands of the L ^A T _E X language for the creation of mathematical texts.	32
ZHURAVSKA I.M. IoT network based on Bluetooth modules for automatized control power consumption	37
ZAIATS V.M., MAJEWSKI J., MARCINIAK T., RYBYTSKA O.M., ZAIATS M.M. Models and algorithms for processing of fuzzy information.	45
MELNYK D.S., ZDOLBITSKA N.V. Research FIR filters using the LabVIEW environment. Create a mobile application in the Android Studio IDE to diagnose smartphones based on the Android OS.	52
KOLOMOIETS G.P. Software of preparation for the printing of diplomas about higher education by data of the USEDEI with the use of modern Java technologies.	57
KOSTIUCHKO S.M., DATSYUK S.V., NAGORNYUK A.I. Sessions and authorization of internet-store users using PYTHON and DJANGO.	66
LAVRECHUK S.V., ILIUSHYK R.S. Advantages of creating real-time applications based on the asynchronous library of Channels and the framework of Django.	70
LYSENKO S.M., HAIBURA Y.O., STETSIUK V.M. The method for Trojan programmes detection based on the fuzzy clustering algorithm	75
MELNIK V.M., N. BAHNIUK., NAGORNYUK A.I. Work features with databases in the Django framework.	83
MELNYK V.M., MELNYK K.V., SHULGA B.V. A comparison of the three most popular web-frameworks for Python project development	88
MELNYK K.V., MELNYK V.M., MATSIBORA A.S. Exploration of spam detection content methods.	95
MISKEVYCH O. I., SYCHOV D. I., KHRISTINETS A.O. On modernization of the local-computer network at PrAT "Volynoblenergo" on the basis of the GRE-tunnel using IPsec encryption.	100
MULIAR V. P. Fundamentals of development of applications with use JavaFX technology.	104
PEKH PETRO, KOBUK ROMAN. Software C++ Builder software=demonstration complex to study equations with parameters by the cross-section	111
PEKH P.A., TYKHOMYROV V.V. Methods for forming system requirements to AS for large 3D programs.	120
POPOV D., KOSTENKO V., YADROVA M., KONDRATYEV S. Development of a multisensory sonar for the blind based on the STM8L microcontroller.	126
SAVENKO O.S., PhD, GRIBINCHUK V.I., KULCHYTSKY M.O. The architecture of the multi-level software system for detecting malware in local computer networks	132
STRUZIK V.A., HRYBKOV S.V., LYTVYN A.O. Study of the methods and approaches of databases refactoring	141
KHRYSTYNETS N.A., BAHNIUK N.V., MATEYCHUK N.V., DZYUBINSKY V.V. Analysis of PHP Templates and Using Twig in Drupal 8.	146
CHORNOBAY K. U., HRYBKOV S.V. Development of budget biometric identification system using neural networks.	151

YATSIUK SVITLANA, YATSIUK ANDREW, MULYAR VADIM. The use of frameworks for the development of web-applications in the language of PHP.	156
<i>AUTOMATION AND MANAGEMENT</i>	
ANDRUSHCHAK I.Ye., MATTHEW Yu.Ya., KOSHELYUK V.A., SIVAKOVSKAYA O.M., MARTSENYUK V.P. Definition of interaction at the same time attended tracks on the module of the vector of the transition.	160
BOMBA A.Y., FEDONIUK A. A. On the generalization of one model of information struggle.	165
GINAYLO P. LISKOVEC S. TYMOSHCHOOK V., GRINCHENKO L. Necessary conditions for extremum with linear multivalued mappings	171
ZHURILO A. G., SIVAK E. M. Some questions about the inconsistency of standards of ESKD and ISO.	176
ZAKHAROVA L.M. Development of an algorithm for dissipative structures identification and determination of their patterns.	181
ISMAILOVA N.P., TRUSHKOV G.V. Geometric modeling of the spatial parametric kinematic screw.	187
KYRYCHENKO ANDRIY, AL IBRAHIMI METAK M.A. Workspace and singularities of 5-axis parallel robot with redundant actuation.	196
KIZYM S.O., ROIKO O.O., YUKHTA O. A. Information and communication technologies in independent work of students	201
KOTSIUBA A.Yu, SITOVSKIY V.O. Realization of the theory of cellular automata for pixel graphics processing.	207
KRAINYK Y.M., PEROV V.O. Organization of Finite-State Machine for Decoding Process Control in Turbo-Product-Code Decoder based on FPGA.	213
KUZMICH OI PH.D., MARKINA L.M., YAKYMCHUK N.M. Modeling and development of a mixing process control system based on the process of water and heat treatment in the production of alcohol.	220
W.W. LYSHUK, Y.R. SELEPYNA, L.W. JASHCHYNISKYY, S.M. KOSTIUTCHKO. Application of the Euler method in dynamics problems.	227
LISKOVEC S., GUDA O. Professional direction of "Higher Mathematics" course for students of economic profile	232
MATVIYIV Y.Ya., ANDRUSHCHAK I.Ye., GANULICH B.K., KRADINOVA T.A. Practical application of the criterion for determining the long-term strength and residual durability of structural elements for low-temperature creep.	238
MOLNAR A.A., BAN H.Y., GAL D.L., GERASIMOV V.V. Telemetry system with elements of augmented reality for the employees of emergency services based on FPGA.	244
POLISHCHUK M.M., GRINYUK S.V., BILOUS M.V. The laboratory block is on the basis of the microcontroller STM32F103VET6.	251
POPLAVSKAYA G.V. Directions of introduction of new information technologies in the process of control and evaluation of students' knowledge.	257
ROYKO L.L., ROYKO O.O. Applied orientation of the course "Mathematics for economists and economic modeling".	263
FILIPCHUK VL, FILIPCHUK LV, SHATALOV AS Safety of operation of automated treatment plants for wastewater treatment with toxic redox impurities.	269
CHERNIKOV M.M., KUVAIIEVA V.I., BOLTENKOV V.A. System of distributed collective expertise with protection from falsifications.	276
O.A. YUKHTA, S.O. KIZYM. O.O. ROIKO, O.YU. ROIKO. Online algorithms analysis on the example of the ski rental problem with two options.	281

ВИМОГИ ДО СТРУКТУРИ ТА ОФОРМЛЕННЯ МАТЕРІАЛУ СТАТЕЙ

➤ **Наукова стаття обов'язково повинна мати наступні необхідні елементи:**

- 1) **постановка проблеми** у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями;
 - 2) **аналіз останніх досліджень і публікацій**, в яких започатковано розв'язання даної проблеми і на які спирається автор,
 - 3) **виділення невирішених раніше частин загальної проблеми**, котрим присвячується означена стаття;
 - 4) **формулювання мети дослідження** (постановка завдання);
 - 5) **виклад основного матеріалу дослідження** з повним обґрунтуванням отриманих наукових результатів; **висновки** з даного дослідження, у тому числі з науковою новизною і
 - 6) **перспективи подальших досліджень** у даному напрямку.
- Статтю можна подавати українською, російською або англійською мовами. Вона повинна бути набрана у текстовому редакторі MS WORD 03/07/10 і надрукована на лазерному або струменевому принтері на білих листах формату А4 (297×210 мм). **Нумерацію сторінок** не виконувати. **Обсяг статті** 5-10 сторінок (не менше).
- **Параметри сторінки.** Верхнє, нижнє та праве поле – 1,5 см, лівє – 2 см. Від краю до верхнього колонтитула – 1,25 см, нижнього – 1,25 см.
- **Шанка статті.** Індекс УДК, ініціали та прізвища авторів, назва організації та назва статті набираються з нового рядка шрифтом Time New Roman Cyr розміром 11 пт з одинарним міжрядковим інтервалом та вирівнюються по лівому краю. Назва статті розміщується через один рядок нижче назви організації (розмір шрифту 11 пт з напівжирним виділенням та вирівнюванням по центру).
- **Анотації** (українською, російською та англійською мовами) повинні містити прізвища та ініціали авторів, назву статті та короткий її зміст і розміщуються через один рядок нижче назви статті та набираються з абзацного відступу 1 см шрифтом Time New Roman Cyr розміром 9 пт з одинарним міжрядковим інтервалом і вирівнюються по ширині. Нижче анотацій обов'язково вказуються **ключові слова**.
- **Основний текст** розміщується на через один рядок нижче анотацій, набирається з абзацного відступу 1 см шрифтом Time New Roman розміром 11 пт з одинарним міжрядковим інтервалом та вирівнюється по ширині.
- **Формули** набираються у редакторі формул MS WORD (використовувати шрифти: Symbol, Time New Roman Cyr; розміри шрифтів: звичайний 12 пт, крупний індекс 7 пт, дрібний індекс 5 пт, крупний символ 18 пт, дрібний символ 12 пт). Формула вирівнюється по центру і не повинна займати більше 5/6 ширини рядка.
- **Ілюстрації**, що присутні у статті, необхідно розташовувати у тексті по центру, вирівнюючи підписи по центру (Рис. 1. Назва). Другий екземпляр ілюстрації необхідно подати на окремому листі. Ілюстрації повинні бути чіткими та контрастними.
- **Таблиці** потрібно розташовувати у тексті по центру, причому їх ширина повинна бути на 1 см менша ширини рядка. Над таблицею ставиться її порядковий номер і назва (Таблиця 1. Назва) та вирівнюється по центру.
- **Посилання** на ту чи іншу роботу повинні позначатися в тексті у квадратних дужках за порядковим номером у списку літератури в кінці статті; посилання на джерела статистичних даних обов'язкові; посилання на публікації дослідників обов'язкові; посилання на підручники, навчальні посібники, газети і ненаукові журнали – небажані; посилання на власні публікації допускаються тільки у випадку крайньої необхідності; роботи авторів, на прізвища яких є посилання в тексті, мають бути в списку літератури до цієї статті.
- **Джерела** повинні бути оформленні за ВАК-формою 23 «Приклади оформлення бібліографічного опису у списку джерел, який наводять у дисертації, і списку опублікованих робіт, який наводять в авторефераті», а їх кількість повинна бути не меншою 10 джерел, що подаються загальним списком в кінці рукопису за абеткою через один рядок після тексту статті.
- До статті обов'язково додається дві **рецензії**: зовнішня і внутрішня. Редакція залишає за собою право направляти статті на додаткову рецензію та відхиляти їх в разі відсутності рецензій.
- Стаття обов'язково подається на лазерному диску. Її також можна переслати електронною поштою за адресою: cit@lntu.edu.ua, ekaterinamelnik@gmail.com.
- В кінці статті обов'язково вказуються ПІБ, посада, науковий ступінь, вчене звання рецензентів статті.
- Рукописи, що не відповідають вище вказаним вимогам, не розглядаються і до друку не приймаються.

ЗРАЗОК ОФОРМЛЕННЯ СТАТТІ

(поля верхнє, нижнє та праве -1.5 см, лівє 2см)

УДК 371.13.001.76

Ткачук Н.М., к.п.н.

Волинський інститут післядипломної педагогічної освіти

ПІДГОТОВКА ВЧИТЕЛІВ ПРОФІЛЬНОЇ ШКОЛИ ДО ВПРОВАДЖЕННЯ ІННОВАЦІЙНИХ ПЕДАГОГІЧНИХ ТЕХНОЛОГІЙ

Ткачук Н.М. Підготовка вчителів профільної школи до впровадження інноваційних педагогічних технологій. У статті розкрито важливість застосування інноваційних педагогічних технологій для реалізації завдань старшої профільної школи. Запропоновано можливості впровадження технологій у навчальний процес шляхом підготовки педагогів в системі післядипломної освіти.

Ключові слова: інновація, технологія, педагогічна технологія, профільна школа, післядипломна педагогічна освіта, підготовка педагога.

Ткачук Н.М. Подготовка учителей профильной школы к внедрению инновационных педагогических технологий. В статье раскрыта важность применения инновационных педагогических технологий для реализации задач старшей профильной школы. Предложены возможности внедрения технологий в учебный процесс в системе последипломного образования.

Times New Roman 9

Ключевые слова: инновация, технология, педагогическая технология, профильная школа, последипломное педагогическое образование, подготовка педагога.

Tkachuk N.M. Teacher training specialized schools to implement innovative pedagogical technologies. The article considers the importance of the application of innovative pedagogical technologies for implementation of tasks of the senior professional schools. The proposed introduction of technology into the learning process through the training of teachers in the system of postgraduate education

Keywords: innovation, technology, educational technology, professional school, post-graduate pedagogical education, training teacher.

Постановка наукової проблеми. Впровадження профільного навчання старшокласників як одного з етапів реформування освітньої галузі зумовлює пошук нових підходів до організації навчання в старшій школі.

.....
.....
.....

Аналіз досліджень. Питанням підготовки вчителів до професійної діяльності в умовах профільного навчання присвятили свої дослідження

.....
.....
.....

Виклад основного матеріалу й обґрунтування отриманих результатів.

Times New Roman 11

ніж розглянути питання підготовки педагогів в процесі підвищення кваліфікації до впровадження інноваційних педагогічних технологій, уточнимо ключові поняття «інновація» та «педагогічна технологія».

.....
.....
.....

Висновки та перспективи подальшого дослідження. Таким чином, одним з найважливіших стратегічних завдань на сьогоднішньому етапі модернізації післядипломної педагогічної освіти є забезпечення якості підготовки фахівців на рівні міжнародних стандартів.

.....
.....

1. Величко Л. Л., Савчин М. М. Теоретичні аспекти педагогічної майстерності та методичної культури педагога [Текст] / Л. Л. Величко, М. М. Савчин // Педагогіка і психологія. – 2013. – № 4. – С. 74–82.
2. Дичківська І. М. Інноваційні педагогічні технології : навч. посіб. / І. М. Дичківська. – К. : Агенція з розвитку освіти, 2010. – 120 с.
3. Липова Л., Войцехівський М., Малишев В. Післядипломна освіта в контексті профілізації педагогічної освіти / Л. Липова, М. Войцехівський, В. Малишев // Післядипломна освіта в Україні. – 2010. – № 1. – С.3–6

Times New Roman 9

Довідки з питань публікації та прийому матеріалів у науковий журнал «Комп'ютерно-інтегровані технології: освіта, наука, виробництво» можна отримати у редактора-коректора – Мельник Катерини Вікторівни за тел. (0332) 74-61-15, або (050)-916-00-81.

Адреса: 43018, м. Луцьк, вул. Львівська, 75, ауд. 141

Автор статті отримує 1 примірник збірника. Вартість друку однієї сторінки становить 40 – грн.(для працівників Луцького НТУ), 50 грн – для інших ВУЗів.

Кошти можна перерахувати на рахунок:

Луцький національний технічний університет
43018 м.Луцьк, вул..Львівська, 75
р/р 31251222217820
банкДКСУ м.Київ,
МФО 820172
Код ЄДРПОУ 05477296

З призначенням платежу: «За інформаційно-аналітичне забезпечення видання «Комп'ютерно-інтегровані технології: освіта, наука, виробництво» від _____(ПІБ)»

Міністерство освіти і науки
Луцький національний технічний університет

Колектив авторів

**КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ:
ОСВІТА, НАУКА, ВИРОБНИЦТВО**

Науковий журнал

Підп. до друку 14.04.2018. Формат А4. Папір офс.
Гарн. Таймс. Ум. друк. арк. 15.25 Обл. – вид. арк. 15.75
Тираж 200 прим. Зам. № 285

Комп'ютерний набір та верстка:

К.В. Мельник,
К.А. Свиридюк.

Редактори:

Ю.О. Мельник,
О.С. Гордіюк.

Друк РВВ Луцького НТУ
Свідоцтво Держкомтелерадіо України ДК №4123 від 28.07.2011 р.
43018, м. Луцьк, вул. Львівська, 75, тел.: (0332) 74-61-02
e-mail: rvv_lntu@ukr.net