

*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

№26 2017

*м. Луцьк
Видавництво Луцького національного технічного університету*

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,

ekaterinamelnik@gmail.com

сайт журналу: **ki.lutsk-ntu.com.ua**

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

№26 2017р.

Журнал засновано у грудні 2010 р.
Свідоцтво про реєстрацію KB № 16705–5277 Р.
Засновник: Луцький національний технічний університет
Рекомендовано до друку Вченою радою
Луцького національного технічного університету
(протокол №9 засідання від 25.04.2017)
Журнал рішенням МОН України
наказом №515 від 16.05.2016р,
включено в перелік наукових фахових видань.
Журнал має російський індекс наукового цитування
(РІНЦ)

ISSN 2524-0560 (Online)
ISSN 2524-0552 (Print)

ЗМІСТ

ІНФОРМАТИКА ТА ОБЧИСЛЮВАЛЬНА ТЕХНІКА

Андрущак І.Е., Андрущук І.В. Концепція і технологія безпеки комп'ютерної мережі.	5
Багнюк Н.В., Мельник В.М., Казмірчук О.В. Аналіз та порівняння можливостей розробки веб-ресурсів засобами PHP-фреймворку Laravel і CMS WordPress.	10
Багнюк Н.В., Мельник К.В., Михайлуца П.С. Сучасні технології реалізації динамічного дизайну web сторінок.	16
Безкоровайна Ю.М., Скалова В.А. Огляд чинних стандартів для розробки захищених інформаційних систем.	22
Бортник К.Я., Зух О.М., Лобода Р.В. Соціальна авторизація.	27
Васильєва Л.В., Синюк Б.В. Дослідження ступеня безпеки високотехнологічних 3D-принтерів.	31
Верещага В.М., Адоньєв Є.О. Композиційний метод утворення Б-поверхонь.	36
Гальчук Т.Н. Комп'ютерна обробка статичних даних контролю точності виробів машинобудування.	42
Горбатюк Р.М. Педагогічні аспекти підготовки інженерів-педагогів засобами інформаційно-комунікаційних технологій.	48
Горшунов А.І. Архітектурна модель побудови веб-застосунків CQRS.	53
Гринюк С.В., Мельник В.М., Равенець А.В. Булеві операції над полігонами при створенні мапи шляхів у тривимірному просторі для комп'ютерних ігор.	58
Гринюк С.В., Поліщук М.М., Грабовський Б.М. Проектування та розробка програмної системи для зберігання і оперування даними про надані послуги і продані товари засобами C++ та QT.	64
Губаль Г.М. Система LATEX у створенні комп'ютерних математичних текстів.	69
Гулай О.І. Методика оцінювання навчальних досягнень студентів.	74
Дудка О.М., Власій О.О. Особливості вивчення програмування на Scratch.	81
Жигаревич О.К., Медведєв М.В. Інформаційна ситема "Студент - ФКНІТ" засобами PHP.	88
Журило А. Г., Сівак Є. М. Деякі питання щодо креслення кіл при побудові аксонометричних проєкцій.	93
Каганюк О.К., Сітовський В.О. Розробка крос-платформних додатків для мобільних пристойів.	98
Коцюба А.Ю., Саковський С.В. Про імпортування баз даних в Android-додаток.	106
Красиленко В.Г., Нікітович Д.В. Моделювання багатокрокових та багатоступеневих протоколів узгодження секретних матричних ключів.	111
Лавренчук С.В., Грідін О.О. Інформаційна система для дипломників та випускників кафедри.	121
Мельник В. М., Гаджамура А. М. Переваги використання СУБД SQLite у додатках Windows.	126
Мельник В.М., Мельник К.В., Багнюк Н.В., Гринюк С.В. Модель генерації вбудованого Андроїд-коду для гетерогенної архітектури.	131
Міскевич О. І., Кокоша М. С. Додаток «Системний монітор» засобами бібліотеки Qt.	138
Міскевич О. І., Коренюк А.Ю. Роль людини в інформаційних технологіях.	143
Пастернак Я.М., Братанов А.С. Порівняльний аналіз типових систем керування контентом (CMS).	147
Пех П. А., Баландюх В. В. Розробка інтернет-магазину з продажу косметики та засобів гігієни на основі CMS WordPress.	154
Ходаков В.Е, Козел В.Н. Оценка эффективности информационных систем по частным показателям качества.	159

АВТОМАТИКА ТА УПРАВЛІННЯ

Гоян Д.І., Краснокутська І.В. Система автоматизації обліку шведсько-українського медичного центру Angelholm.	164
Жигаревич О.К., Котлярець В.В., Лець А.Р. Модель екосистеми навчального програмного забезпечення.	168
Здолбіцька Н.В., Здолбіцький А.П. Пристрій підвищення енергоефективності системи керування електромагнітними клапанами.	178
Здолбіцька Н.В., Здолбіцький А.П., Калінін Б.Ю. Комп'ютерна система візуальних ефектів.	183
Каганюк О.К. Аналіз систем автоматичного керування вугільно-добувним комбайном в складних гірничо-геологічних умовах.	188
Коцюба А.Ю., Мацібора А.С. Інформаційна система аналізу постачальників сировини на підприємство.	194
Мусієнко М.П., Журавська І.М. Система контролю безпілотних апаратів, заснована на використанні мобільних пристроїв.	199
Орлова М.М., Мосійчук І.В. Модифікований алгоритм пошуку найкоротших маршрутів передачі інформації в одорангових мережах.	204
Поліщук М.М., Данилюк Л.М., Буденко Ю.В. Електронний термометр із голосовим повідомленням температури.	208
Поліщук М.М., Каганюк О.К., Сопіжук Р.В. Дослідження DC-AC перетворювач з мікроконтролерним керуванням частоти інвертора.	213
Поліщук М.М., Гринюк С.В., Дмитерчук В.В. Робототехнічна система автоматичного пошуку джерела світла.	220
Решетило А.О., Решетило О.М., Гуменюк П.О., Маркіна Л.М. Системи забезпечення мікроклімату салону легкового автомобіля.	227
Шабелюк О.В. Розробка бібліотеки фізичного обладнання для побудови навчальної системи на основі доповненої реальності.	233

УДК 61:004.651 (075.8)

І.Є.Андрушак, І.В.Андрошук

Луцький національний технічний університет

CONCEPT AND SECURITY TECHNOLOGY COMPUTER NETWORKS

I.Ye.Andrushchak, I.V.Androshchuk. Concept and security technology computer networks. In this paper, a review of the history of the formation of the issue of technology and the concept of security of computer networks, current state and prospects of development of methods and tools for security in networks. Highlight key aspects of building planes and the use of technology in modern security systems of computer networks. Highlighted key issues, normative and technical security of computer networks.

Keywords: Computer network, content filtering, information security, integrated content filtering system, model receiving content.

І.Є.Андрушак, І.В.Андрошук. Концепція та технологія безпеки комп'ютерної мережі. В роботі проведено огляд історії становлення питання про технологію та концепцію безпеки комп'ютерних мереж, сучасного стану та перспектив розвитку методів та засобів безпеки в мережах. Виділено ключові площини аспектів розбудови та застосування технологій систем безпеки в сучасних комп'ютерних мережах. Підкреслено основні проблемні питання, нормативно- правового та технічного характеру безпеки комп'ютерних мереж.

Ключові слова: комп'ютерна мережа, фільтрація контенту, інформаційна безпека, комплексні системи фільтрації контенту.

И.Е.Андрушак, И.В.Андрошук. Концепция и технология безопасности компьютерной сети. В работе проведен обзор истории становления вопрос о технологии и концепции безопасности компьютерных сетей, современного состояния и перспектив развития методов и средств безопасности в сетях. Выделены ключевые плоскости аспектов развития и применения технологий систем безопасности в современных компьютерных сетях. Подчеркнуто основные проблемные вопросы, нормативно- правового и технического характера безопасности компьютерных сетей.

Ключевые слова: компьютерная сеть, фильтрация контента, информационная безопасность, комплексные системы фильтрации контента, модель получения контента.

Security Information Network Protection includes hardware, software, data and personnel. Network security consists of the provisions and policies adopted by the network administrator to prevent and monitor unauthorized access, misuse, modification or denial of the computer network and available resources. Network security includes permission to access the data network that is provided by the network administrator. Users choose or are assigned their ID and password or other authentication information, allowing them to access the information and applications within their authority.

Network security includes a variety of computer networks, both public and private, which are used in the daily workplace for transactions and relationships between enterprises, government agencies and individuals. Networks can be private, such as inside a company or open to the public. Network security is involved in organizations, enterprises and other types of institutions. The most common and easy way to protect network resources is to give them a unique name and corresponding password [1].

At a time when computer systems are the foundation of security of entire countries and databases - the main capital of many companies, virus protection stands next to general issues of economic security as the country as a whole and individual institutions and organizations. Especially this problem for institutions and businesses that deal with sensitive data about customers. Theft, destruction, distortion of information, failure and rejection computer systems - the problems that carry the viruses and virus programs.

Threats to information security can be divided into two types: threats that are random (their source may be unintentional errors in software, failure of hardware, incorrect user actions) and threats intended that, unlike random, pursuing the goal causing intentional damage information system, data theft, damage to natural or legal persons.

Network security starts with authentication usually involves a username and password. If you need only one piece of authentication (username), it is called single-factor authentication. With two-factor authentication, the user still has to use a security token or 'key' credit card or mobile phone at control authentication, the user must use a fingerprint scan or take retina.

After authentication, firewall provides access to network users. For detection and suppression action malware using antivirus software or intrusion prevention systems (IPS).

The connection between two computers using network can be encrypted to preserve

confidentiality.

Data protection is one of the main problems of computer networks as an advantage network access to shared data and devices, and this leads to the possibility of unauthorized access to data.

Data security is to protect network resources from destruction and protect data from accidental or intentional disclosure, as well as the unlawful changes.

Designed to ensure the security of the network administrator. In large networks this purpose there are special positions (security officers). To ensure data security developed multilevel system of protection:

- built-in protection - software and system (passwords, permissions);
- physical protection - locks, door security, alarm;
- administrative control - arrangements, orders administration;
- legislation and social environment - the laws of copyright and property rights, intolerance of software piracy.

US Department of Defense in the book "Computer Security Evaluation Criteria" (Orange Book), has identified seven levels of security and computer network systems. This development became common in the world to classify the security of the system. , The following levels of protection [3]:

- D – the level of minimum protection (Minimal Protection). Restricted systems at other levels do not guarantee the required level of security;

- C1 – level of protection selective (Discretionary Protection). Allows users to apply access restrictions to protect private information;

- C2 – controlled level of access (Controlled Access Protection). Includes requirements Level C1, and the protection of the registration process in the system records events protection, resource isolation different processes;

- B1 – protection by category (Labeled Protection). By requirements of C2 added ability to protect individual files, records in the files, and other objects of special security tags stored together with these objects. It is believed that such protection can overcome well-trained hacker as a normal user - no;

- B2 – Structured protection level (Structured Protection). By B1 level requirements attached full protection of all system resources directly or indirectly available to the user. It is believed that hackers can not penetrate the system with such protection;

- B3 – Domain level security (Security Domains). For the demands of B2 is added to a clear specification of users who are prohibited access to certain resources, fuller sign of potentially dangerous events. It is believed that even experienced programmers unable to overcome the system with this level of security;

- A1 – verified level of development (Verified Design). Full protection. Specified and verifiable mechanisms for protection. It is believed that in the system with this level of protection without permission can not penetrate no.

Some systems (like banking or tax) need not user identification and individual. Distinguishes several ways such identification.

For personal physical characteristics (biometrics). Remove fingerprint or hand geometry, retina, pupil, facial features, and then analyzed. Alternatively, the system offers again a number of randomly selected words and analyzes the features of voice.

For a subject that person-to-user to carry. This can be the subject of a special badge, magnetic card code. This method is cheap but unreliable, the subject can be faked, steal, etc. [5].

For that person must know or remember. We must remember the password or correctly answer a series of questions. This method is the cheapest and most common, but unreliable (password can be selected answers guess).

Safety is ensured by the use of a network the right to access, authentication and registration of connections.

The identification process is called user authentication. Standard authentication method - using a user name and password as the previous pair of identifiers is intended that the user must enter at the request system for obtaining access to network resources. In this, the most simple form of authentication user ID and password are transmitted over the network in clear text (ie not encrypted). The process of authentication - comparing pairs identifiers transmitted with the records of the table located on the server - is performed according to the protocol for authentication password (Password Authentication Protocol,

PAP). Recordings are stored encrypted, unlike couples identifiers transmitted and this is a weakness of this method of authentication.

A more advanced challenge-response system operates according to the authentication protocol for establishing communication with the (Challenge Handshake Authentication Protocol, CHAP). Under this protocol, agent authentication (software located on the server) sends the user to a key by which that encrypts your username and password and sends this information back to the server. Authorization - the process of providing the user the right access to the system in which a user name and password assigned to him are recorded in a special table systems [7].

Widespread system that provides a high level of protection for authentication, challenge-response system, which uses a smart card.

Control attempts to access the network, you can easily determine whether an unauthorized user was trying to sneak in, and see if you forgot your password anyone of staff.

Blocking access. In many organizations as user IDs cited their initials and surname. Attackers try to sneak in, it was enough to know the following. Software developers have created a program block access to the system. Often software that performs blocking access allows you to specify another threshold, this threshold is determined by the time during which the system will be locked.

An important concept of data protection issues in the networks is recognition.

Recognition - a guarantee that information (package) came from a legitimate source rightful recipient.

Indeed, one of the most common practices malicious networks are packet sniffing and substitute their own or directing them to another destination. Therefore, all modern network protocols are usually equipped with means of recognition. One of the mechanisms of recognition packages are hotel and destination of the same pseudorandom number generator. Each package indicate random number which is compared with the same number of the recipient.

A similar task performs electronic signature - a sequence of bytes that form special algorithms and whose authenticity can be verified.

To recognize use separate servers that issue electronic certificates. Servers used in the certification of sufficiently powerful operating systems.

One of the most prominent solutions is the system of centralized recognition Kerberos (it is implemented by programming and is compatible with all types of systems. System works in client-server paradigm. It consists of client software, deployed on workstations, users, and server applications. There are three types of server programs: recognition server, the server permits and server administration. In the process of recognition of the client involved in the first two of these servers. Each server has a scope defined by the content of its database of users).

To measure recognition accuracy using two indicators: the percentage of false identification (False Acceptance Rate (FAR)) and the percentage of false recognition (False Rejection Rate (FRR)).

The original meaning of the firewall (firewall) - a wall in a building made of fire-resistant and non-flammable material that can prevent the spread of fire. In computer network firewall - a computer with a software system which is installed on the network edge and allow that only authorized packets in a certain way. [6]

Most firewalls to protect internal corporate network from attacks from external networks. However, they can be used to filter the output information, limiting user access internal network outside.

Sometimes firewall functions in complex systems distributed among their own firewalls and servers intermediaries. The firewall protects the network from external influence. It filters link layer frames, recognizes the session that opens external user. Intermediary server controls and limits the output of internal user outside, and often his representative.

Features intermediary server, hiding internal addresses of stations, handing out the entire network as a computer server address; caching popular web-pages, files, so users do not have to turn to the external network. Popular information server updates automatically defined intervals.

Firewalls use different filtering algorithms, they have different degrees of protection and cost. To classify firewalls describe their work using a reference model OSI.

There are:

- Firewalls filtering packets (packet filtering firewall; operate at the data link, network levels);
- Locks the session level (circuit level gateway; working on session level, recognize the session);

- Application level gateways (application level gateway; filtered information for applications);
- Expert-level firewalls (stateful inspection firewall; perform the functions of firewalls all lower levels).

Typically, the higher the level of the firewall, the more protection it provides.

Firewall packet filtering work with hardware or software router. They analyze the content of IP-packet header and based on the information they have and their rules table and decide about the passage of the package or its rejection. Often the information for which decide to pass the package is its complete address information, information about protocol and application port numbers recipient and sender. If the package does not meet any of the rules, the rule "default". It often rejects the package. The specific configuration depends on the rules of politics. Firewalls generate small delay transmitting messages. Often filtering packets in routers integrate. However, the level of protection in such firewalls insignificant - the attacker may substitute the IP address [9].

Gateways of the session participants recognize the session. Verification procedures performed only at the beginning of the session. After the client and server authenticity is confirmed, a gateway simply copies packets without performing filtering. Gateways of the session supported operating table sessions and when the session ends, destroying an entry. Copying packages carry out a special program, called channel-mediators (pipe proxies). Gateways of the session can perform the function of an intermediary server which displays the internal LAN addresses into one (actually for firewall). For packets coming in the opposite direction, reverse operation is performed. Consequently, network address space reserved - external user sees internal addresses. However, such gateways do not provide sufficient protection and is usually not a separate product, supplying them with gateways of applications.

Penetration testing (penetration testing, pentest - tests to overcome the defense) - a detailed analysis of networks and systems in terms of a potential attacker. The essence of the test is sanctioned attempt to circumvent existing remedies complex information system. During testing plays the role of attacker specialist who must determine the level of protection, identify vulnerabilities, identify the most likely way of hacking and determine how well the detection and protection of information systems against attacks on the company. Penetration test allows you to get an objective assessment of how easily access corporate network resources and the company's website, how and through which vulnerability. Penetration test is to simulate the actions of the attacker on the entry into the information system and allows you to identify most vulnerabilities on the network. Penetration test is carried out to obtain an independent assessment of the security of their corporate networks.

The goal of penetration testing is to identify weaknesses in the protection of IP and, if possible, and corresponds to the wishes of the customer, to make a show evil.

The main objective of penetration testing, simulating the action completely attacker to carry out an attack on the Web server, application server, or database, personnel, corporate network. Penetration testing can be performed as a part of auditing for compliance, as well as independent work.

Penetration test - a method of evaluating the security of a computer system or network by partial simulation for external intruders from penetration into it (which means no authorized access to the system) and internal attackers (who have a certain level of authorized access). This process involves an active analysis of the system to identify any potential vulnerability that may result from improper system configuration, known and unknown defects of hardware and software or operational lag in procedural or technical countermeasures. This test is done from a position of a potential attacker and can involve active use of vulnerabilities.

Security problems that have been identified during the penetration test, presented to the owner of the system. Effective Penetration Test combine this information with accurate assessment of the potential impact on the organization and outline the boundaries of technical and procedural countermeasures to reduce risks.

Testing is about a variation of things in the software and its surroundings, which can be changed by varying them, and seeing how the software responds. The aim is to ensure that the software performs safely and securely at reasonable or even unreasonable production scenarios. Thus, the most fundamental is planning a test that can be done to understand what can be changed and how that change should be organized to test. In terms of safety, the environment, user input, and internal data and logic are the main places where such changes might reveal safety problems, which consist of files, applications, system resources, and other local or network resources. Any of them can be the starting point of attack. User

input that comes from outside (usually unreliable) persons analyzed and used software. Internal data and logic stored internally as well as variables and logical ways that a number of potential transfers.

Penetration test is useful for several reasons:

- determine the possibility of a particular set of attacks;
- identify vulnerabilities of highest risk resulting from the combination of lower risk vulnerabilities used in sequence.

By applying special programs correspond intermediary. They can perform filtering level applications. Each application can have its mediator. In contrast agents in the session-level gateways, intermediaries of applications analyze packets at applications. For example, the mediator can use FTP to ban the use of teams put to prohibit transfer of information to your server [7].

Firewalls expert level combine features of all previous systems. They perform packet filtering link-level gateways recognize the session as the session level and are able to analyze and filter packets on the basis of applications. Unlike firewalls of applications that actually transmit information between two transmission links dissolved client-gateway and gateway external computer and cause a significant delay in transmitting information, expert level firewalls establishing a direct link between the identified client and server. For flow filtration using custom templates, heuristic rules relative to standards other methods from the arsenal of expert systems. Firewalls expert level provide the highest level of protection and higher performance options.

Protecting the network using firewalls. The firewall is usually set between the router and the network that protects, and a computer with two network adapters. One adapter is connected to the hub so-called demilitarized network (DMZ), the other - to the hub network, which is protected. A firewall typically connect to through it passed all traffic "Internet - a network that protects". It is important to note that since access to DMZ-concentrator are just a router and firewall, all data exchange with the Internet is through the firewall [10].

The proxy service is a mediator between the host, the host service, and service and most used protocols such as FTP, Telnet. The firewall processes connection requests, meaning that it functions as a proxy service. Many proxy service allow you to use FTP or disable specific FTP-command.

Firewall software is made, check the contents of the package, perform proxy services, encryption, authentication and generate alerts. To check suspicious traffic (such as repeated attempts to connect to the network) analyzes the contents of packets with the same IP-address of the destination. Further action depends on the firewall configuration, or deny any further suspicious packages or situation reported firewall administrator.

1. Simmonds. A; Sandilands. P; van Ekert. L (2004). An Ontology for Network Security Attacks. Lecture Notes in Computer Science. Lecture Notes in Computer Science 3285. c. 317–323. doi:10.1007/978-3-540-30176-9_41. ISBN 978-3-540-23659-7.
2. Role-Based Trusted Network Provides Pervasive Security and Compliance - interview with Jayshree Ullal, senior VP of Cisco.
3. Dave Dittrich, Network monitoring/Intrusion Detection Systems (IDS), University of Washington.
4. Гайкович В. Ю. Необхідна умова їх успішного застосування / В.Ю.Гайкович, Д.В. Єршов // "Системи безпеки, зв'язку та телекомунікацій", №3, – 2007.
5. Єршов Д.В., Попова З.В. Навчання комп'ютерної безпеки / Д.В. Єршов, З.В. Попова // «Ком.Лог», №2, – 2007.
6. Кадыров М.М. Информационная безопасность компьютерных сетей / М.М. Кадыров, Н.О. Каримова // Молодой ученый. – 2016. – №8. – С. 124-126.
7. Лукацкий А.В. Нові підходи до забезпечення інформаційної безпеки мережі / А.В. Лукацкий // Комп'ютер-Прес. – № 7. – 2007.
8. Лукацкий А.В. . Адаптивна безпеку. Данина моді чи усвідомлена необхідність / А.В. Лукацкий // PCWeek, – №37. – 2009.
9. Лукацкий А.В. Сімейство засобів адаптивного управління безпекою / А.В. Лукацкий // «Мережі», – № 10. – 2008.
10. Новіков С.М. Захист інформації в мережах зв'язку з гарантованою якістю обслуговування. / С.М.Новіков // Новосибірськ. – 2003. – С.18-31.

УДК 004.023

Багнюк Н.В. к.т.н, доцент, Мельник В.М. к.т.н, доцент, Казмірчук О.В., магістрант
Луцький національний технічний університет

АНАЛІЗ ТА ПОРІВНЯННЯ МОЖЛИВОСТЕЙ РОЗРОБКИ ВЕБ-РЕСУРСІВ ЗАСОБАМИ PHP-ФРЕЙМВОРКУ LARAVEL І CMS WORDPRESS

Багнюк Н.В., Мельник В.М., Казмірчук О.В. Аналіз та порівняння можливостей розробки веб-ресурсів засобами PHP-фреймворку Laravel і CMS WordPress. У статті розглядається аналіз та порівняння двох веб-сайтів, розроблених на CMS WordPress та PHP фреймворку Laravel. Основні критерії порівняння: швидкодія завантаження сторінок, швидкодія вивантаження оголошень, час розробки веб-сайту, структура системи, захищеність системи.

Ключові слова: PHP, CMS, Laravel, WordPress, фреймворк, JavaScript, CSS, HTML.

Багнюк Н.В., Мельник В.М., Казмірчук О.В. Анализ и сравнение возможностей разработки веб-ресурсов средствами PHP-фреймворка Laravel и CMS WordPress. В статье рассматривается анализ и сравнение двух веб-сайтов, разработанных на CMS WordPress и PHP фреймворка Laravel. Основные критерии сравнения: быстродействие загрузки страниц, быстродействие выгрузки объявлений, время разработки веб-сайта, структура системы, защищенность системы.

Ключевые слова: PHP, CMS, Laravel, WordPress, фреймворк, JavaScript, CSS, HTML.

Bahniuk N.V., Melnyk V.M., Kazmirchuk O.V. Analysis and comparison capabilities for developing web resources means PHP-Laravel framework and CMS WordPress. In the article the analysis and comparison of two websites developed on CMS WordPress and PHP framework Laravel. The main criteria for comparison, the speed of page loading, unloading performance ads, a website development structure system, security system.

Keywords: PHP, CMS, Laravel, WordPress, framework, JavaScript, CSS, HTML.

Вступ.

На сьогоднішній день майже кожний заклад, організація чи підприємство мають свій сайт. Для їх розробки застосовується велика кількість засобів, починаючи з мов програмування, наприклад, PHP, C#, Perl, Ruby, Python і закінчуючи фреймворками та системами керування вмістом.

PHP є однією з найбільш популярних мов програмування, а також недавній випуск PHP 7 [1] зробив його кращим і стабільнішим на стороні сервера, ніж будь-коли. PHP широко використовується у великих проектах. Facebook, наприклад, використовує PHP для збереження і створення своїх внутрішніх систем. WordPress використовує PHP, щоб оживити свої внутрішні інформаційно-трансляційні проблеми, які в свою чергу надають руху більш ніж 26% мережі. На сьогоднішній день на PHP працюють понад 82% веб-сайтів.

У цій статті ми оприлюднимо порівняння трьох найбільш популярних PHP фреймворків: Symfony, Laravel і Yii [2,3]. Ми розглянемо і порівняємо їх, щоб допомогти вирішити, який з них буде найкращим чином підходити під потреби розробника.

Аналіз існуючих способів.

Навіщо потрібен PHP фреймворк взагалі? Який сенс використовувати фреймворк замість чистого PHP для розробки програми? Наведемо кілька переваг використання фреймворків:

- PHP фреймворк робить розробку швидше. Наприклад, розробнику не потрібно писати складні запити для отримання даних з бази. Фреймворки включають CRUD операції (Create, Read, Update і Delete).
- Фреймворки дозволяють розробникам легко масштабувати системи.
- Код додатка короткий і з ним легко працювати.
- MVC забезпечує швидку розробку.
- Безпека веб-додатків є помітно вищою.
- Мінімальна кількість коду має максимальний ефект.
- Вищевказані переваги є надто значними, щоб бути зігнорованими.

Незважаючи на це, чистий PHP може бути використаний для створення будь-якої програми, якщо діючі стандарти вимагають розробки інструментів і навичок тайм-менеджменту для задоволення ринкового попиту. Все ж Laravel має відмінне співтовариство і випереджає всі в списку найбільш популярні фреймворки. Одним з провідних розробників Laravel на Liveness.tv є Sfikell. У травні 2015 року Laravel оголосила, що версія 5.1 буде включати довгострокову підтримку протягом двох років. Версія 5.2 була представлена в грудні 2015 року. Багато хостинг

компанії надають підтримку Laravel і пропонують хостинг рішення для додатків на Laravel. Laravel також корисна, але вони підтримує менше, ніж бази даних Symfony. Підтримка баз даних цим фреймворком наступна: LQLite, MySQL, PostgreSQL, Redis, Microsoft BI, Mongo DB.

Однак якщо в цілому порівнювати Symfony, Laravel і Yii, то всі три PHP фреймворки є відмінними варіантами, які забезпечують full-stack розробку. Але в багатьох типах додатків Laravel є переможцем, який перетворюється в зірку.

Останнім часом стали також дуже популярні системи CMS – система управління вмістом або система управління контентом (англ. Content management system) — комп'ютерна програма, що використовується для управління вмістом чого-небудь (звичайно цей вміст розглядається як неструктуровані дані наочної задачі в протилежність структурованим даним, звичайно що знаходяться під управлінням СУБД (Системи Управління Базами Даних)). Звичайно такі системи використовуються для зберігання і публікації великої кількості документів, зображень, музики або відео. [4,5] Нижче в алфавітному порядку представлені приклади CMS. Наявність в цьому списку не вказує на популярність системи: Drupal, Etomite CMS, eZ publish, Irokez, Joomla, Mambo, moJoPortal, WordPress. CMS рішень багато, але при виборі варто спиратися на наступні можливості:

- призначений для користувача сервіс – наявність тих або інших функцій і модулів, зрозумілість і доступність користувачу;
- технологічність – використання технологій, що дозволяють підвищити пропускну спроможність і надійність системи;
- сумісність (апаратна і програмна) – можливість функціонування системи на різних платформах, сумісність з СУБД, можливість підключення додаткових модулів;
- масштабованість – можливість розвитку і нарощування системи.

WordPress WordPress – це проста у встановленні та використанні система керування вмістом з відкритим кодом, яка широко використовується для створення веб-сайтів. Сфера застосування – від блогів до складних веб-сайтів. Вбудована система тем і плагінів в поєднанні з вдалою архітектурою дозволяє конструювати на основі WordPress практично будь-які веб-проекти, [6] написані на мові програмування PHP з використанням бази даних MySQL. Сирцевий код поширюється на умовах ліцензії GNU General Public License Управління системою та інші можливості WordPress Дизайн:

- простота встановлення, простота налаштувань;
- підтримка веб-стандартів (XHTML, CSS);
- модулі для підключення (плагіни) з унікально простою системою їх взаємодії з кодом; можливість автоматичного встановлення та оновлення версії безпосередньо з панелі адміністратора;
- підтримка так званих «тем», з допомогою яких легко змінюється як зовнішній вигляд, так і способи виведення даних;
- можливість редагувати шаблони одразу в панелі адміністратора;
- «теми» реалізовані як набори файлів-шаблонів на PHP (у HTML-розмітку вставляються PHP-мітки);
- багато бібліотек «тем» і «плагінів»;
- потенціал архітектури дозволяє легко реалізовувати складні рішення;
- SEO-оптимізована система;
- наявність українського перекладу.
- миттєва публікація;
- наявність ЛЗУ (людино-зрозумілий URL);
- редагування WYSIWYG-редактором з можливістю вставлення форматowanego тексту (наприклад з програми Microsoft Word) або редагування за допомогою HTML-розмітки.

Плагіни Wordpress – це елементи для розширення функціональності CMS [7]. Одне встановлення WordPress дозволяє вести одночасно лише один блог. Якщо ж є бажання створити і вести на одному сервері декілька блогів, то можна встановити потрібну кількість WordPress в різні каталоги (віртуальні сервери) і в СКБД створити окрему базу даних для кожного блогу. Гілка WordPress Multi-User (WordPress MU, чи WPMU) дає змогу створити декілька блогів в одній інсталяції. WordPress MU також дозволяє кожному користувачу системи вести довільну кількість блогів і контролювати їх з панелі управління. Для кожного блогу створюється 8 таблиць в базі

даних. Ще одна перевага у використанні платформи WordPress полягає у вмонтованій підтримці пошукової оптимізації (SEO) – незамінною складовою для підвищення рейтингу сайту в пошукових системах Google, Yandex та інші.

У даній роботі розглянуті системи на мові PHP, такі як система керування вмістом WordPress та фреймворк Laravel (рис.1). Особливість WordPress полягає в тому, що це готова система, на яку існує багато готових шаблонів та плагінів. Розробка базових сайтів без унікальних особливостей значно швидша, проте швидкодія досить низька. Тоді як розробка на Laravel значно зручніша, система має більшу швидкодію, хоча розробка займає більше часу.

Постановка наукової проблеми. Мета даної роботи – розробити два веб-сайти на CMS WordPress та PHP фреймворку Laravel та порівняти процес їх розробки та роботи.

У даній роботі розглянуті системи на мові PHP, такі як система керування вмістом WordPress та фреймворк Laravel (рис.1). Особливість WordPress полягає в тому, що це готова система, на яку існує багато готових шаблонів та плагінів. Розробка базових сайтів без унікальних особливостей значно швидша, проте швидкодія досить низька. Тоді як розробка на Laravel значно зручніша, система має більшу швидкодію, хоча розробка займає більше часу.



Рис. 1. Логотипи Laravel та Wordpress

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Як зазначено в меті роботи, було розроблено два веб-сайти. Перший – для продажу автомобілів. Для реалізації даних продажів найкраще підходить веб-сайт типу «дошка оголошень». На сайтах такого типу кожен користувач може створити оголошення для продажу свого авто. Будь-хто може переглянути інформацію про оголошення та побачити контактні дані продавця. Дана система буде зручна як для звичайної людини, яка хоче продати свій автомобіль, так і для компаній, які продають їх масово.

Для зручної роботи було добавлено багато функцій для створення і роботи з оголошеннями.

Для початку повинні бути записи у такому форматі, як нам потрібно, а саме, мати різноманітні поля для опису даної машини, наприклад: марка, тип кузова, тип палива, колір, ціна, стан і інше.

Доступ до створення оголошень мають всі зареєстровані користувачі, просто натиснувши кнопку «Добавити оголошення». Після чого відбувається перехід в адміністративну частину, де проходить створення оголошення. Для того, щоб оголошення відображалось на різних мовах, пропонується створити варіант опису та назви для кожного.

Будь-який товар може бути добавлений до списку улюблених, для того щоб потім знайти товар, який вам сподобався без докладання зусиль.

Знайти ваш новий автомобіль

В даний час ми маємо 16 оголошень доступних на нашому складі.

Виберіть стан

Тип автомобіля

Марка

Модель

Місто

Область

Рік з)

Рік до)

Ціна з (\$)

Ціна до (\$)

пошук

Рис. 2. Фільтр пошуку оголошень

Також для швидкого пошуку потрібного транспорту існує система, яка дозволяє відфільтрувати за: станом, типом автомобіля, маркою, моделлю, містом, областю, роком випуску, ціною.

У базі є практично усі моделі автомобілів та мотоциклів, актуальні на даний час. Окрім того, на сторінці «Мої записи» можна керувати своїми оголошеннями.

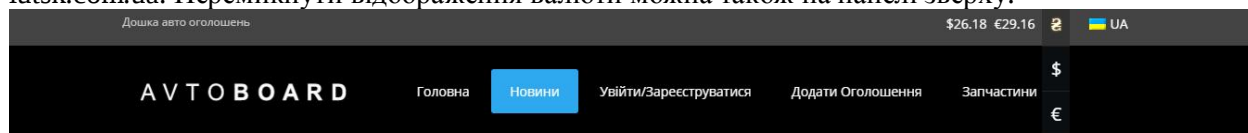
	2010 BMW 335i xDrive Curabitur scelerisque, nibh at cursus luctus, ipsum neque dapibus dolor, blandit hendrerit purus ante eu tortor. Mauris hendrerit ultricies pellentesque. Nam laoreet commodo metus, sed...	publish	
	1966 Cadillac Sedan de Ville Curabitur scelerisque, nibh at cursus luctus, ipsum neque dapibus dolor, blandit hendrerit purus ante eu tortor. Mauris hendrerit ultricies pellentesque. Nam laoreet commodo metus, sed...	publish	
	2008 Porsche Cayenne Curabitur scelerisque, nibh at cursus luctus, ipsum neque dapibus dolor, blandit hendrerit purus ante eu tortor. Mauris hendrerit ultricies pellentesque. Nam laoreet commodo metus, sed...	publish	
	2010 Mercedes-Benz SLR Curabitur scelerisque, nibh at cursus luctus, ipsum neque dapibus dolor, blandit hendrerit purus ante eu tortor. Mauris hendrerit ultricies pellentesque. Nam laoreet commodo metus, sed...	publish	
	2008 Range Rover Sport Curabitur scelerisque, nibh at cursus luctus, ipsum neque dapibus dolor, blandit hendrerit purus ante eu tortor. Mauris hendrerit ultricies pellentesque. Nam laoreet commodo metus, sed...	publish	
	2007 Audi S4 Curabitur scelerisque, nibh at cursus luctus, ipsum neque dapibus dolor, blandit hendrerit purus ante eu tortor. Mauris hendrerit ultricies pellentesque. Nam laoreet commodo metus, sed...	publish	

Рис. 3. Вивід власних оголошень

При редагуванні запису можна переходити до адміністративної частини (схожої як для додавання), де є можливість змінити будь-які його дані.

На сайті наявна мультимовність з підтримкою української, російської та англійської мов. Елемент перемикачів між ними знаходиться на панелі вгорі.

Усі ціни в оголошеннях зазначаються в доларах, але для зручності сприйняття можна ввімкнути відображення цін у інших валютах. Курс валют постійно оновлюється за даними піко-lutsk.com.ua. Перемикнути відображення валюти можна також на панелі зверху.



Записи

Головна > listings > ЛуАЗ 969 Волинь



Рис. 4. Шапка сайту

Розміщення оголошень на даній дошці є безкоштовним, хоча якщо потрібно швидше продати авто, то є можливість підняти його в рейтингу. Для цього розроблений плагін, який може надавати більший пріоритет оголошення на певний час та надавати стан «терміново» чи «гарячий».

Другий веб-сайт розроблений на фреймворці Laravel як система для створення та розповсюдження «квестів». Під квестами розуміється будь-яка подія, розписана покроково, із можливістю залучення до неї людей.

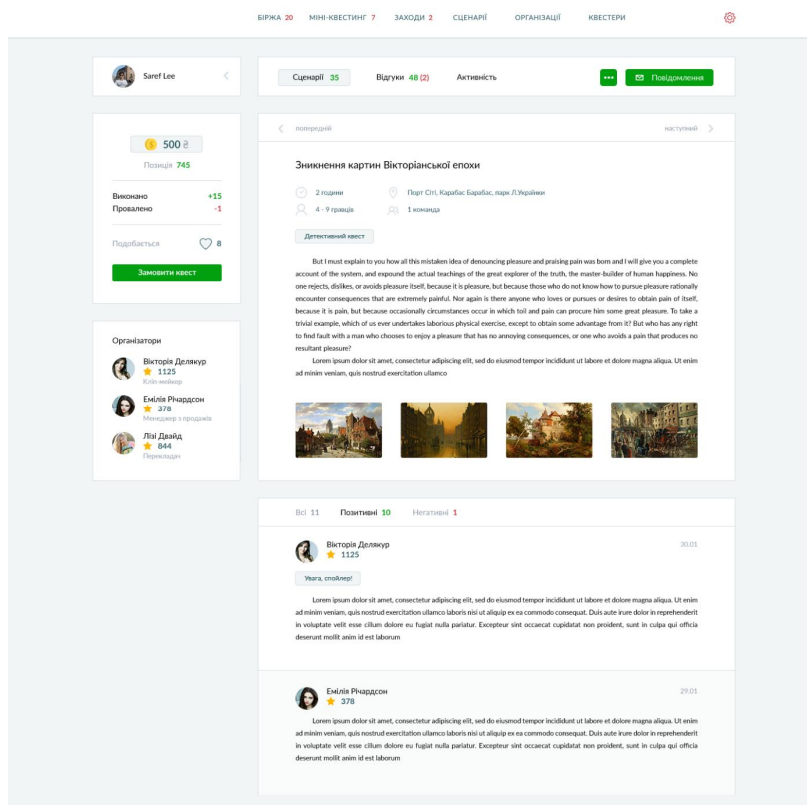


Рис. 5. Сторінка виводу сценарію

Наступним кроком роботи є розробка редактора квестів, що дозволить спланувати його покроково та призначити виконавців кожного з них. Також у майбутньому буде розроблено мобільний додаток, що дозволить редагувати, керувати та відслідковувати події для учасників. Для зручності відслідковування буде інтегрована карта, на якій вказуватиметься маршрут до даної цілі.

Даний веб-сайт виконано у вигляді соціальної мережі. Користувачі можуть створювати як власні квести, так і глобальні, за додаткову плату. Окрім цього замовник квеста зможе найняти виконавців, наприклад, розіграти друга на день народження.

Висновки

У даній роботі проаналізовано та порівняно можливості розробки веб-ресурсів засобами php-фреймворку Laravel і CMS WordPress на прикладі двох сайтів.

1. PHP 7.0.6 Released. <http://php.net/archive/2016.php#id2016-04-29-1>.
2. М. J. Garbade. Как выбрать PHP фреймворк? <https://habrahabr.ru/post/305626/>.
3. PHP-Дайджест № 100 – интересные новости, материалы и инструменты (1 – 15 января 2017). <https://habrahabr.ru/company/zfort/blog/319580/>.
4. Горнаков С. Г., Осваиваем популярные системы управления сайтом (CMS). – М.: ДМК Пресс, 2009. – 336 с.
5. Басюк Т.М. Принципи побудови системи аналізу та просування інтернет-ресурсів / Т.М. Басюк // Вісник Нац. ун-ту "Львівська політехніка" "Комп'ютерні науки та інформаційні технології". – 2012. – № 784. – С.43–48.
6. Офіційний сайт WordPress. – Режим доступу : <https://ru.wordpress.org/>.
7. Wordpress [електронний ресурс] // Режим доступу: <http://mywordpress.ru/about/> – Назва з екрану.
8. Востров А. Как появился Wordpress [Електронний ресурс] / Алексей Востров – Режим доступу до ресурсу: <http://www.seoded.ru/istoriya/internet-history/wordpress.html>.
9. Емельянов А. Г., Талалаев А. А., Тищенко И. П., Фраленко В. П. Нейросетевая технология обнаружения сетевых атак на информационные ресурсы // Программные системы: теория и приложения: электрон. научн. журн. 2011. № 3(7), С. 3–15.
10. Крейн Д. AJAX в действии: технология / Д. Крейн, Э. Паскарелло, Д. Даррен. – М.: Вильямс, 2010. – 640 с.
11. Ларсен Р. Введение в AJAX [Електронний ресурс] / Роб Ларсен. – 2012. – Режим доступу до ресурсу: <https://www.ibm.com/developerworks/ru/library/wa-aj-ajaxhistory/>.
12. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. / Робин Никсон. – Санкт-Петербург: Питер, 2015. – 688 с. – (3).
13. Правильный Переклад Wordpress Або Навіщо потрібні Файли Мо I По [Електронний ресурс] – Режим доступу до ресурсу: <http://ukrarticles.pp.ua/pk-internet/6507-pravilnyj-perevod-wordpress-ili-zachem-nuzhny-fajly-mo-i-po.html>.
14. Роли пользователей WordPress и их права доступа [Електронний ресурс]. – 2012. – Режим доступу до ресурсу: <http://fairheart.ru/wordpress/dopolnitelnye-plaginy/roli-polzovateley-wordpress.html>.
15. Флэнаган Д. JavaScript. Подробное руководство / Д. Флэнаган. – Санкт-Петербург: Символ-Плюс, 2013. – 1080 с.
16. Холзнер С. Ајах. Библия программиста / Стивен Холзнер. – М.: Вильямс, 2009. – 560 с. – (Библия пользователя).
17. Chakraborty S. A Guide to WordPress Custom Post Types: Creation, Display and Meta Boxes [Електронний ресурс] / Soumitra Chakraborty. – 2012. – Режим доступу до ресурсу: <http://code.tutsplus.com/tutorials/a-guide-to-wordpress-custom-post-types-creation-display-and-meta-boxes--wp-27645>.
18. Getting Started with WordPress [Електронний ресурс] – Режим доступу до ресурсу: https://codex.wordpress.org/Getting_Started_with_WordPress.
19. Schifreen R. The Web Book: The ultimate beginner's guide to HTML, CSS, JavaScript, PHP and MySQL / Robert Schifreen., 2009. – 472 с. – ISBN-10/ASIN: B007LTM67L.
20. Spannagle P. WordPress and Flash 10x Cookbook / P. Spannagle, S. Soward., 2010. – 268 с. – ISBN 139781847198822.
21. Theme Development [Електронний ресурс] – Режим доступу до ресурсу: https://codex.wordpress.org/Theme_Development.
22. Writing a Plugin [Електронний ресурс] – Режим доступу до ресурсу: https://codex.wordpress.org/Writing_a_Plugin.

УДК 004.023

Багнюк Н.В. к.т.н, доцент, Мельник К.В. к.т.н, доцент, Михайлуца П.С., студент
Луцький національний технічний університет

СУЧАСНІ ТЕХНОЛОГІЇ РЕАЛІЗАЦІЇ ДИНАМІЧНОГО ДИЗАЙНУ WEB СТОРІНОК

Багнюк Н.В., Мельник К.В., Михайлуца П.С. Сучасні технології реалізації динамічного дизайну web сторінок. У статті проведено огляд еволюції принципів побудови web сторінок, а також пропонується спосіб їх розробки з використанням скриптових мов програмування та дослідження принципу реалізації динамічної зміни властивостей різноманітних об'єктів в процесі її активної експлуатації.

Ключові слова: динамічний, дизайн, parallax, web сторінки, cms, framework, javascript.

Багнюк Н.В., Мельник К.В., Михайлуца П.С. Современные технологии реализации динамического дизайна web страниц. В статье проведен обзор эволюции принципов построения web страниц, а также предлагается способ их разработки с использованием скриптовых языков программирования и исследование принципа реализации динамического изменения свойств различных объектов в процессе ее активной эксплуатации.

Ключевые слова: динамический, дизайн, parallax, web страницы, cms, framework, javascript.

Bahniuk N.V., Melnyk K.V., Mykhailutsa P.S. Modern technologies of dynamic web page design. This article provides an overview of the evolution of the principles of building web pages, and provides a method of development using scripting languages, as well as research principles of the dynamic changes in the properties of different objects during its active operation.

Keywords: translation, dynamic, design, parallax, web pages, cms, framework, javascript.

Вступ. Сучасні рішення реалізації веб сторінок та загалом веб ресурсів базуються на використанні складного набору різноманітних інструментів та засобів розробки. Розвиток технології звичайної передачі гіпертексту та реалізації його на стороні клієнта притерпівся суттєвих змін за останні 30 років. Ринок послуг, які повинні задовольняти усім вимогам до наповнення сторінки також повинен відповідати вимогам щодо сучасних технологічних рішень.

Реалізація динамічного дизайну сторінки висуває низку вимог:

- використання скриптів в процесі експлуатації сторінки;
- зчитування поточного стану сторінки, положення мишки, прогортання;
- внесення змін «на льоту» в код сторінки або її параметри, наприклад, стиль;
- дизайн повинен пригортати увагу користувача.

Більшість сучасних рішень побудовані на основі JavaScript, що обґрунтовано його гнучкістю та набором модулів для роботи з складовими частинами сторінки та легкістю маніпуляції DOM об'єктами.

Постановка наукової проблеми. Метою даної роботи є аналіз існуючих способів розробки динамічного web дизайну в розрізі еволюції принципів розробки з використанням скриптових мов програмування та дослідження реалізації принципу динамічної зміни властивостей різноманітних об'єктів web сторінки в процесі її активної експлуатації.

Аналіз еволюції принципів розробки web сторінок та існуючих засобів.

Для розуміння принципу побудови динамічного дизайну web сторінки слід відмітити чотири основних етапи технологічного розвитку виконання сторінок, що відрізняються основним набором інструментів.

Перший етап характеризувався статичними сторінками, які базувались на гіпертекстовій розмітці сторінки (HTML, XHTML, DHTML) (рис. 1). Використання таких технологій з плином часу вимагало залучення великих ресурсів для створення сторінок. Чим складніша сторінка (наповнена великою кількістю даних), тим більше часу необхідно було затрачати для її побудови та обслуговування.

```
1      <!DOCTYPE html>
2      <html lang="en">
3      <head>
4          <meta charset="UTF-8">
5          <title>Title</title>
6      </head>
7      <body>
8          <h1>Hello World!</h1>
9      </body>
10     </html>
```

Рис. 1. Приклад гіпертекстової розмітки сторінки

На підтримку розробникам сторінок було запропоновано використання динамічно-генерованого вмісту сторінок та використання технології збереження даних у базах даних, що спростило керування такими ресурсами, а процес написання нових сторінок супроводжувався реалізацією алгоритмів, які забезпечували збір та розміщення даних на сторінці з використанням алгоритмів та функцій мов програмування. Залучення на другому етапі таких мов, як наприклад, PHP змінили підхід до створення сторінок загалом. Основну вимогу до знання чи використання гіпертекстової розмітки не було відмінено, її розширили, давши можливість модульного розширення функціоналу [1].

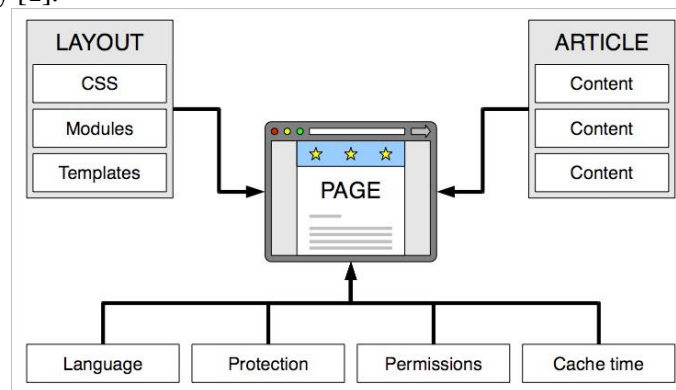


Рис. 2. Структура сторінки, згенерованої системою управління вмістом

Використання мов програмування дало можливість ще більш спростити процес розробки об'ємних ресурсів в мережі Інтернет та використати повноцінну модель «клієнт-сервер», що в свою чергу дало старт третьому етапу еволюції, а саме – використанню систем управління вмістом CMS та системам, побудованих на базі Framework технології (робочих середовищ розробки). Такі системи надають набір спеціалізованого або базового функціоналу для побудови профільних, унікальних ресурсів. Управління вмістом повністю покладається на ядро такої системи і в залежності від її типу або спеціалізації потребує набору інструментів для його роботи. Наприклад, однією з найпопулярніших сьогодні систем є WordPress, що побудована на основі PHP, може використовувати велику кількість модулів з набором різноманітних функцій, або JavaScript, система AngularJS, яка є також модульною, але являє собою звичайний набір файлів, реалізованих на мові JavaScript. Структуру CMS показано на рисунку 2. Для прикладу, серед середовищ можна взяти сучасний YII 2, що побудована на базі PHP або середовища мови Python: Django, Flask, Tornado. Структуру веб додатків, побудованих на основі Framework, показано на рисунку 3. Як CMS, так і Framework системи генерують вміст сторінки для відображення її у браузері клієнта ресурсу, основна ж їх відмінність є лише місце генерації цього вмісту – сервер або ж клієнтський браузер[2].

Виділяють також четвертий етап розвитку, який характеризується запуском нової версії гіпертекстової мови розмітки сторінок HTML5. Загалом, дана нова «мова» є гібридом мови програмування та звичайної розмітки, що дозволяє викинути зі схеми додаткові інструменти,

наприклад Flash додатки, які несуть за собою низку проблем з безпекою та споживають велику кількість ресурсів на стороні клієнта для їх відображення. Можемо спостерігати поступове залучення такої технології побудови ресурсів, наприклад, на ресурсі YouTube, де успішно використовується HTML відео програвач. Великі компанії такі, як Amazon, Google, Cisco також розпочали перехід веб інтерфейсів на дану мову. Наприклад, Amazon впровадив новий інтерфейс управління масивами серверів, основна маса коду якого реалізована лише з використанням HTML5.

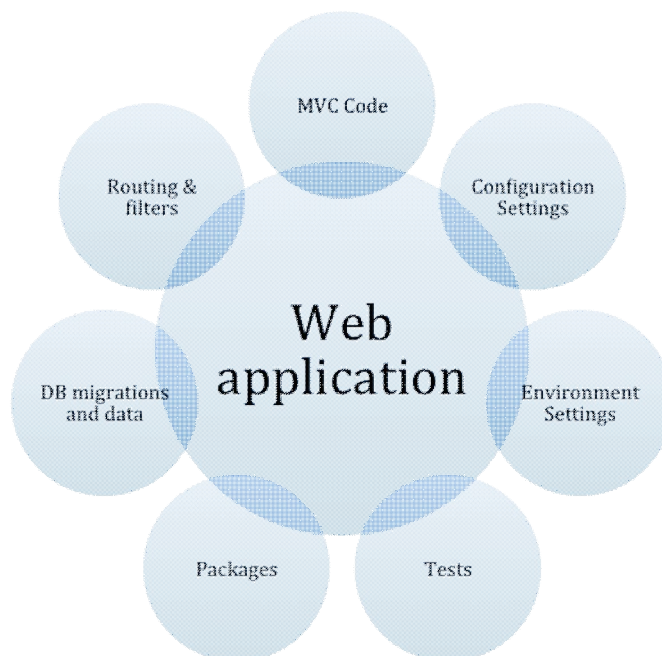


Рис. 3. Основні складові Web додатків, побудованих на базі Framework

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Для побудови динамічного дизайну сторінки використовується скриптовий код, що реагуватиме на зміну параметрів на сторінці або в процесі перегляду сторінки. З розвитком технологій та поширенням мобільних платформ (планшетів та сучасних телефонів), дуже актуальних є зміна вмісту або стану сторінки саме в процесі прогортання її, так званого «скролу», що дозволить пригорнути більшу увагу до самої сторінки. В галузі електронного маркетингу – пригортання уваги до продукту є одним з основних засобів ведення успішного бізнесу. Для реалізації цієї технології пропонується залучити технологію Parallax. Структура побудови зображення використовує багатоплановий, принцип як показано на рисунку 4.

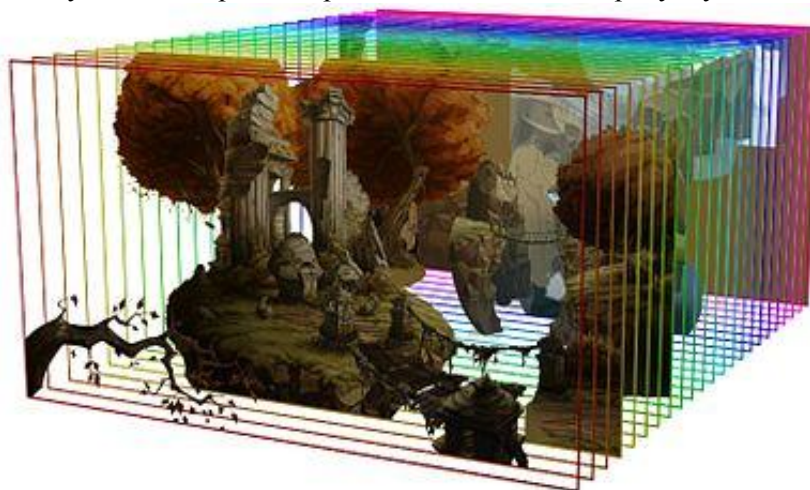


Рис. 4. Шари зображення Parallax

Кожному з шарів зображення можна задати свою динаміку та напрямок руху. Для акцентування уваги на окремих елементах сторінки використовується ефект впливання елементів сторінки в процесі її до низу або до верху. Кожен з шарів рухається з своєю швидкістю та змінюється як показано на рисунку 5.

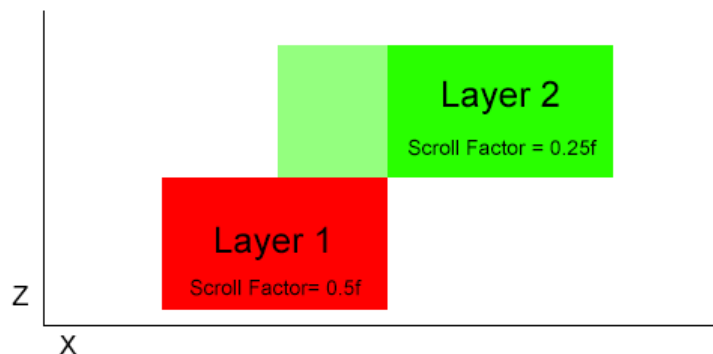


Рис. 5. Нашарування об'єктів на сторінці

Для реалізації таких ефектів використовується принцип ігор початку 90-х для ігрових консолей того часу. Найпростішим варіантом є використання зображень, як шпалер на сторінці, кожне з яких буде знаходитись на своєму шарі зі своїм стилем CSS. Рух таких шарів виконує алгоритм, що розміщується в скрипті для цієї сторінки. Наприклад, для реалізації 4 слайдів для прогортання на сторінці можна створити блок з 4 секцій в HTML розмітці сторінки:

```
<main>
  <section id="slide-1" class="homeSlide">
    <div class="bcg">
      <div class="hsContainer">
        <div class="hsContent">
          <h2>Найпростіший спосіб пригорнути увагу є...</h2>
        </div>
      </div>
    </div>
  </section>
  <section id="slide-2" class="homeSlide">
    <div class="bcg">
      <div class="hsContainer">
        <div class="hsContent">
          <h2>використати динамічний дизайн за допомогою...</h2>
        </div>
      </div>
    </div>
  </section>
  <section id="slide-3" class="homeSlide">
    <div class="bcg">
      <div class="hsContainer">
        <div class="hsContent">
          <h2>використання Parallax ефекту лише...</h2>
        </div>
      </div>
    </div>
  </section>
  <section id="slide-4" class="homeSlide">
    <div class="bcg">
      <div class="hsContainer">
        <div class="hsContent">
          <h2>додавши його на сторінку.</h2>
        </div>
      </div>
    </div>
  </section>
```

</main>

Кожному з блоків, як вказано у розмітці, задано свій клас та унікальну назву для спрощення стилізації їх у розмітці даної сторінки, наприклад, для визначення положення першої секції можна задати:

```
<section id="slide-1" class="homeSlide">
  <div class="bcg"
    data-center="background-position: 50% 0px;"
    data-top-bottom="background-position: 50% -100px;"
    data-anchor-target="#slide-1">
    <div class="hsContainer">
      <div class="hsContent"
        data-center="bottom: 200px; opacity: 1"
        data-top="bottom: 1200px; opacity: 0"
        data-anchor-target="#slide-1 h2">
        <h2> Найпростіший спосіб пригорнути увагу є...</h2>
      </div>
    </div>
  </div>
</section>
```

Як видно з коду, положення задається абсолютними значення в пікселях для початкового розташування та позиції зміни за допомогою data-attributes. Для управління динамічною передачею стану та зміни положення слід використати скрипт. Для отримання параметрів data-attributes, можна їх зчитати з документу за допомогою скрипта[3]:

```
/^data(?:-?(_\w+))?(?:-?(?:\d*\.\d+p)?)?(?:-?(start|end|top|center|bottom))?(?:-?(top|center|bottom))?$;/
```

За допомогою параметрів data-center та data-top-bottom браузер отримує директиву на зсув позиції центру картинки першого слайду на 100 пікселів, доки не перейде до низу першого слайду. Перехід на наступний слайд відбудеться при досягненні параметру data-top за допомогою якоря data-anchor-target, що в свою чергу вказує на об'єкт, який має з'явитись.

Динаміку прогортання можна задати набором функцій, визначивши початкову позицію, як нуль, та кінцеву, як одиницю, проміжні ланки задати звичайним математичним розрахунком наприклад[4]:

```
begin: function() {return 0;};
end: function() {return 1;};
```

Для лінійної прокрутки:

```
linear: function(p) {return p;};
```

Для прокрутки в квадратній прогресії:

```
quadratic: function(p) {return p * p;};
```

Для прокрутки в кубічній прогресії:

```
cubic: function(p) {return p * p * p;};
```

Для повороту:

```
swing: function(p) {return (-Math.cos(p * Math.PI) / 2) + 0.5;};
```

Інші типи прокрутки:

```
sqrt: function(p) {return Math.sqrt(p);};
```

```
outCubic: function(p) {return (Math.pow((p - 1), 3) + 1);};
```

```
bounce: function(p) {
  var a;
  if(p <= 0.5083) {
    a = 3;
  } else if(p <= 0.8489) {
    a = 9;
  } else if(p <= 0.96208) {
```

```
    a = 27;  
  } else if(p <= 0.99981) {  
    a = 91;  
  } else {  
    return 1;  
  }  
  return 1 - Math.abs(3 * Math.cos(p * a * 1.028) / a);  
}
```

де p – множник заданої прокрутки.

Завдяки даному підходу можна реалізувати різноманітні ефекти появи об'єктів на сторінці та їх зникнення. Приведемо приклад горизонтальної прокрутки, де за допомогою не складних математичних перетворень положення об'єкту на сторінці, його кут повороту та швидкість появи можна змінювати за потреби у будь-якому положенні у трьохвимірному просторі.

Висновки

Запропонований метод створення динамічного дизайну для сайту є дуже популярним для невеликих проектів та великих компаній. Загальний об'єм даних, які підлягають трансформації на сторінках, проаналізованих у дослідженнях для цієї статті, напряму залежить від складності їх реалізації. Існують окремі проекти, які повністю базуються на технології Parallax та використовують велику частину її можливостей. Більшість існуючих рішень залучають динаміку зміни об'єктів на сторінці лише частково.

Дана технологія є чудовим рішенням для сайтів електронної комерції, фотоальбомів, одно-сторінкових ресурсів. Нажаль, технологію Parallax не об'єктивно використовувати в різноманітних інформаційно-навантажених системах, наприклад, інформаційних порталах чи панелях управління. Основним її недоліком є складність розрахунку позицій об'єктів та їх зміни при побудові їх на базі CMS або ж Framework. Цікавим рішенням для останніх платформ може стати інтеграція таких ефектів до структури розрахунків серверної частини фреймворку, що може в рази спростити процес генерації сторінок.

Існуючі скрипти, реалізовані на мові JavaScript для роботи з технологією Parallax, дозволяють її інтеграцію у більшість систем управління вмістом, що спрощує процес реалізації динамічного дизайну на різнотипних проектах.

1. Jon Duckett "HTML and CSS: Design and Build Websites" - John Wiley & Sons, U.S.A, том 9, №3, Nov 2011.
2. Dr. Andy Williams "Wordpress for Beginners 2017: A Visual Step-by-Step Guide to Mastering Wordpress", Kindle Edition, Dec 2016.
3. Douglas Crockford "JavaScript: The Good Parts" – Wilmington, DE, U.S.A May 2014
4. Ved Antani. " Mastering JavaScript". Packt Publishing Ltd. Birmingham UK, Jan 2016.

УДК 004.056

Безкоровайна Ю.М., старший викладач, Скалова В.А., старший викладач
Національний авіаційний університет

ОГЛЯД ЧИННИХ СТАНДАРТІВ ДЛЯ РОЗРОБКИ ЗАХИЩЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Безкоровайна Ю.М., Скалова В.А. Огляд чинних стандартів для розробки захищених інформаційних систем. У статті розглядається чинні стандарти для розробки захищених інформаційних систем на основі стандарту, який об'єднує чинні стандарти та заходи, які пропонуються реалізувати під час розробки захищених інформаційних систем.

Ключові слова: безпека, стандарти, розробка програмне забезпечення, інформаційні системи, захист.

Безкоровайная Ю.Н., Скалова В.А. Обзор действующих стандартов для разработки защищенных информационных систем. В статье рассматриваются действующие стандарты для разработки защищенных информационных систем на основе стандарта, который объединяет действующие стандарты и меры, которые предлагаются реализовать при разработке защищенных информационных систем.

Ключевые слова: безопасность, стандарты, разработка программного обеспечения, информационные системы, защита.

Bezkorovaina Y.M., Skalova V.A. The review of existing standards for the development of secure information systems. The article examines the current standards for the development of secure information systems based on a standard that combines existing standards and measures that are proposed to implement in the development of secure information systems.

Keywords: security, standards, software development, information system, security.

Вступ.

На сьогоднішній день гострою проблемою стоїть питання про реалізацію та визначення захищених інформаційних систем (ІС). Однією з проблем забезпечення захисту інформації це розробка безпечного програмного забезпечення (ПЗ), які входять до його складу. Одним із важливих питань при розробці безпечного ПЗ - є питання вчасного впровадження та початок робіт з реалізації заходів з безпеки та результати цих заходів. Закордоном швидкими темпами розробляють та впроваджують стандарти та нормативні документи з розробки безпечного ПЗ. В статті буде розглянуто заходи та їх результат, які відповідають стандартам та нормативним документам, які чинні в Україні.

Аналіз останніх досліджень і публікацій. Виділення невирішених раніше частин загальної проблеми.

Невпинне розширення сфери застосування засобів комп'ютерної обробки інформації і комп'ютерних засобів телекомунікації залучають до сфери інформаційних технологій все більше коло людей, що підвищує ризики виникнення інформаційних загроз та їх реалізації. В 2016 закордоном був затверджений та введено в дію стандарт з розробки безпечного ПЗ, який об'єднує загальні вимоги при розробці безпечного ПЗ [4], на його основі було проаналізовано заходи та їх вплив на реалізацію безпечного ПЗ, апарат комплексів заходів [1, 2, 7]. Проте виникає проблема - цей стандарт не чинний на території України, хоча включає посилання на чинні стандарти та нормативні документи. В статті приводяться заходи та пов'язані з ними стандарти та нормативні документи.

Формулювання мети дослідження.

З огляду на зростання складності інформаційних систем актуальними стають проблеми загрози безпеки інформації, пов'язані з наявністю вразливості ПЗ, що використовуються в складі ІС. Для захисту від загроз, як правило, використовують комплекс заходів, які реалізовані в процесах експлуатації та супроводу ПЗ. Проте для забезпечення необхідного рівня захисту інформації потрібна реалізація заходів, які спрямовані на запобігання появи і усунення вразливостей ПЗ в процесах життєвого циклу (ЖЦ) ПЗ, пов'язаних з проектуванням, реалізацією і тестуванням [4].

Сучасний підхід до захисту інформації в ІС ґрунтується на ідеальних моделях процесу організації захищених систем та їх функціонування. Ідеальна модель передбачає комплексний підхід до створення безпечних ІС протягом всього життєвого циклу таких систем. У той же час відсутність загальної теорії захисту інформації призводить до того, що при створенні систем інформаційного захисту рішення щодо способів реалізації їх елементів приймаються на основі

евристичних моделей або експертних оцінок існуючих варіантів реалізацій систем захисту інформації [10].

Актуальними проблемами захисту інформації на сьогодні є: розробка елементів загальної теорії захисту інформації в захищених системах; створення та дослідження універсальних моделей безпечних ІС; створення та дослідження моделей загроз ІС; розробка методів і процедур об'єктивного аналізу ефективності прийнятих рішень при створенні систем інформаційної безпеки [10].

Виклад основної частини.

Безпечне ПЗ - це ПЗ, яке розроблене з використанням сукупності заходів, які направлені на попередження появи та усунення вразливості програми [4], тобто недоліків ПЗ, які використовується для реалізації вразливості безпеки інформації.

Розробка ПЗ поділяється на процеси ЖЦ ПЗ, в якому виділяють основні, допоміжні та організаційні процеси ЖЦ ПЗ за стандартом ДСТУ ISO 12207 [5]. До основних відносяться: аналіз вимог до ПЗ, проектування архітектури ПЗ, конструювання ПЗ та тестування ПЗ; до допоміжних та організаційних: інсталяція та впровадження, менеджмент документації та конфігурації ПЗ, інфраструктури середовища розробки та людських ресурсів [5].

За стандартом [4] розробка безпечного ПЗ поділяється на дев'ять заходів на основі процесів ЖЦ ПЗ. На рисунку 1 зображено взаємозв'язок між процесами ЖЦ ПЗ та заходи, які необхідно документувати за стандартами.



Рис. 1. Взаємозв'язок між процесами ЖЦ ПЗ та стандартами

Аналіз вимог до ПЗ. При аналізі вимог до ПЗ розробник повинен визначити вимоги до безпеки, які висуваються до ПЗ. Для реалізації необхідно створити документи за стандартами ГОСТ 19.201-78 «ЕСПД. Технічне завдання» [3, с. 43-44; 8] та відповідно з вимогами сімейства ASE «Оцінка завдання з безпеки» за стандартом ДСТУ ISO/IEC 15408 [6].

В результаті вдалої реалізації заходів формується перелік вимог до безпеки, що пред'являється до ПЗ та враховується при проектуванні архітектури ПЗ.

Проектування архітектури ПЗ. При проектуванні архітектури ПЗ реалізовується моделювання загрози безпеки інформації та уточнюється проекту архітектури ПЗ з урахуванням

результатів моделювання. Це сприяє створенню та документуванню проекту архітектури ПЗ з урахуванням вимог з безпеки, для його подальшого використання в процесах ЖЦ ПЗ, пов'язаних з реалізацією та тестуванням ПЗ; формування вихідних даних для виконання динамічного аналізу коду програми, тестування на проникнення в рамках процесу кваліфікаційного тестування ПЗ.

Проект архітектури ПЗ розробляється з урахуванням необхідності виконання вимог з безпеки; формують перелік виявлених потенційних загроз безпеки інформації, які використовуються при проведенні динамічного аналізу коду програми та тестування на проникнення на етапі тестування. Розробник ПЗ повинен виконувати моделювання загроз безпеки інформації, які можуть виникнути в наслідок використання ПЗ.

Для організації робіт необхідно документально оформити проект архітектури ПЗ за стандартами ГОСТ 19.402-78 «Опис програми» [3, с. 51-52; 8] та ГОСТ 19.404-79 «Пояснювальна записка. Вимоги до вмісту та оформлення» [3, с. 56-57; 8] та відповідно стандарту ДСТУ ISO/IEC 15408 [6] вимог сімейства:

- AOV.FSP «Функціональна специфікація»;
- AOV.TOS «Проект ОО»;
- AOV.ARC «Архітектура безпеки».

Конструювання ПЗ. При конструюванні ПЗ необхідно реалізувати наступне: використовувати при розробці ПЗ ліцензійні інструментальні засоби, створити вихідний код на основі уточненого проекту архітектури ПЗ з дотриманням порядку оформлення вихідного коду, який містить перелік правил та рекомендацій; провести статичний аналіз та експертизу вихідного коду, виявити та усунути недоліки (вразливих конструкцій) в вихідному коді, в іншому випадку документувати обґрунтування факту в кожному випадку відмови від використання у вигляді коментарів в вихідному коді; задокументувати вихідні дані для виконання динамічного аналізу коду та тестування на проникнення в рамках процесу тестування ПЗ; забезпечувати проведення статичного аналізу та експертизу вихідного коду з ціллю виявлення недоліків програми в вихідному коді, який необхідно проводити по відношенню до компонентів, які запозичені у сторонніх розробників ПЗ, якщо для цих компонентів доступний вихідний код програми.

За результатами статичного аналізу та експертизи вихідного коду програми можна проводити доопрацювання програми.

Для організації робіт документацію розробника ПЗ необхідно розроблювати у відповідності стандарту ДСТУ ISO/IEC 15408 [6] вимог сімейства:

- ALC.TAT «Інструментальні засоби і методи»;
- ALC_CMC «Можливості управління конфігурацією»;
- ALC.CMS «Область управління конфігурацією».

Тестування ПЗ. При тестуванні ПЗ розробник ПЗ необхідно реалізовувати наступні заходи: функціональне тестування програми; тестування на проникнення; динамічний аналіз коду; фаззінг тестування.

Документація розробника ПЗ повинна містити: план тестування; опис виконуваних тестів та інструментальних засобів, які використовують для тестування; фактичні результати тестування; звіти, які містять списки виявлених невідповідностей вимогам безпеки та вразливості; опис дій, які направлені на їх усунення, обґрунтування неможливості або відсутності необхідності в усуненні невідповідностей вимогам та вразливості, виявлені при моделюванні дій потенційних порушників; проекту архітектури ПЗ, в тому числі інформації про запозичені у сторонніх розробників ПЗ компоненти; результати моделювання загроз безпеки інформації; результати тестування на проникнення; результатів динамічного аналізу вихідного коду.

За результатами тестування можна проводити доопрацювання, яке направлено на усунення виявлених невідповідностей вимогам безпеки та/або вразливості.

Документацію розробника ПЗ необхідно розробляти у відповідності стандарту ДСТУ ISO/IEC 15408 [6] вимог сімейства:

- ATE_COV «Покриття»;
- ATE_OPT «Глибина»;
- ATE_FUN «Функціональне тестування»;
- AVA_VAN «Аналіз вразливостей».

Інсталяція та підтримка ПЗ. При інсталяції та підтримки ПЗ необхідно реалізувати наступні заходи: забезпечити захист ПЗ від загроз безпеки інформації, пов'язаних з порушенням цілісності, в процесі його передачі користувачу; поставка користувачеві експлуатаційну документацію. В склад ПЗ, яке постачається, повинні бути включені експлуатаційні документи в об'ємі, достатньому для правильного налаштування та безпечного використання програми.

Документація розробника ПЗ повинна містити опис виконаних технічних та організаційних заходів, які використовуються для визначення модифікацій ПЗ або будь-якого розходження між оригіналом та версією, отриманою користувачем — розроблюється експлуатаційні документи відповідно вимогам ГОСТ 19.101-78. «ЕСПД. Види програм та програмних документів» [3, с. 18-20; 8] та відповідно стандарту ДСТУ ISO/IEC 15408 [6] вимог сімейства:

- ALC_DEL «Постачання»;
- ACD.OPE «Керівництво користувача з експлуатації»;
- AGO.PRE «Підготовчі процедури».

Експлуатація ПЗ. В процесі експлуатації ПЗ розробник повинен реалізовувати наступні заходи: реалізація та використання процедури відстеження та виправлення виявлених помилок ПЗ та вразливостей програми; систематичний пошук вразливості програми. При реалізації даних заходів необхідно визначити причини помилок ПЗ та вразливості програми та прийняти заходи з попередження подібних помилок ПЗ та вразливостей програми в майбутньому. Процедура виправлення помилок ПЗ та вразливостей програми повинна забезпечити прийом та обробку повідомлень від користувачів про помилки ПЗ та вразливостей програми та запити на їх усунення. За результатами обробки повідомлень від користувачів про помилки ПЗ та вразливостей програми можна проводити доопрацювання програми.

Документація ПЗ повинна містити опис альтернативних способів тимчасових рішень проблеми, пов'язаних з екстреною ситуацією; опис методів доведення до користувачів інформацію про вразливість програми та рекомендації по їх усуненню; опис методів прийому та обробки повідомлень від користувача; опис процедури відстеження та виправлення виявлених помилок ПЗ та його вразливості; перелік екстрених ситуацій, при яких можливий випуск оновлень ПЗ в обхід стандартної процедури випуску нових версій ПЗ; список виявлених помилок ПЗ та вразливість програми та опис дій, які направлені на їх усунення, або обґрунтування неможливості або відсутності необхідності в їх усуненні — розробляти відповідно до стандарту ДСТУ ISO/IEC 15408 [6] вимог сімейства:

- ALC.FLR «Усунення недоліків»;
- ALC.TAT «Інструментальні засоби і методи»;
- ALC.CMC «Можливості управління конфігурацією»;
- ALC.CMS «Область управління конфігурацією».

Менеджмент документації та конфігурації ПЗ. В процесі менеджменту документації та конфігурації ПЗ розробник повинен реалізовувати наступні заходи: реалізація та використання процедури унікального маркування кожної версії ПЗ; використання системи управління конфігурацією ПЗ.

Для організації робіт та підтвердження відповідності вимогам необхідно реалізувати: опис методів, які використовуються для унікального маркування кожної версії ПЗ та ідентифікації елементів конфігурації; перелік елементів конфігурації, які пов'язані з реалізацією функцій безпеки ПЗ та повинні контролюватися системою керування конфігурацією ПЗ; порядок використання системи керування конфігурацією ПЗ з ціллю визначення всіх елементів конфігурації, на які впливає модифікація елемента конфігурації відповідно стандарту ДСТУ ISO 10007:2005 «Системи управління якістю. Настанови щодо управління конфігурацією» [9], розробляти відповідно стандарту ДСТУ ISO/IEC 15408 [6] вимог сімейства:

- ALC.CMC «Можливості управління конфігурацією»;
- ALC.CMS «Область управління конфігурацією».

Менеджмент інфраструктури середовища розробки ПЗ. В процесі менеджменту інфраструктури середовища розробки ПЗ розробник повинен реалізувати наступні заходи: захист від несанкціонованого доступу до елементів конфігурації; резервне копіювання елементів конфігурації; реєстрація подій, які пов'язані з фактами змін елементів конфігурації.

Документація розробника ПЗ повинна містити: відомості про періодичність та перелік елементів конфігурації, які підлягає резервному копіюванню; опис застосованих технічних та організаційних заходів, які забезпечують захист від несанкціонованого доступу до елементів конфігурації, резервне копіювання та відновлення визначених елементів конфігурації, реєстрацію всіх подій, які пов'язані з фактами змінення елементів конфігурації, в журналі реєстрації подій; перелік елементів конфігурації, які повинні бути захищені від загроз безпеки інформації, які пов'язані з порушенням конфіденційності, цілісності та доступності - необхідно розробити у відповідності стандарту ДСТУ ISO/IEC 15408 [6] вимог сімейства:

- ALC_DVS «Безпека розробки»;
- ALC.CMC «Можливості управління конфігурацією».

Менеджмент людськими ресурсами. В процесі менеджменту людськими ресурсами розробник ПЗ повинен реалізовувати наступні заходи: періодичність навчання співробітників; періодичний аналіз програми навчання співробітників.

Для підтвердження відповідності вимогам документація розробника ПЗ повинна містити: правила та програми навчання; відомості про періодичність навчання; відомості про проходження співробітників навчання.

Висновки.

В процесі огляду стандартів було виявлено актуальність стандартів, які чинні на території України [3, 5, 6, 8, 9], відсутність нормативних посилань на таких етапах процесу ЖЦ ПЗ, як менеджмент людськими ресурсами та особливості при розробці захищених ІС.

Напрямок подальшого дослідження є виявлення суміжних стандартів та їх опрацювання для подальшого дослідження проблеми захисту ІС у зв'язку з скасуванням міждержавного стандарту ГОСТ 19 серії (починаючи з 2018 року) [8] та стандартів з менеджменту людськими ресурсами, які не враховані в оглянутому стандарті [4].

1. 28 магических мер безопасного программного обеспечения [Электронный ресурс] : Материалы VI Всероссийской конференции «Безопасные информационные технологии» Вопросы кибербезопасности / Барабанов А.В., Марков А.С., Цирлов В.Л. - 2015 - Выпуск №5(13) - С. 2-10. - Режим доступа: http://cyberrus.com/wp-content/uploads/2015/12/02-10-513-15_1.-Баранов.pdf
2. Борисов С. СОИБ. Анализ. Разработка безопасного ПО [Электронный ресурс]. / С. Борисов - 08.08.2016 р. - Режим доступа до статті: <http://www.securitylab.ru/blog/personal/sborisov/308857.php>
3. Единая система программной документации: сборник / Национальные стандарты. – Изд. официальное. – М. : Стандартинформ изд., 2005. – 128с.
4. Защита информации. Разработка безопасного программного обеспечения. Общие требования : ГОСТ Р 56936-2016. - [Действует от 2016-06-01]. - М.: Стандартинформ, 2016. - 24 с. - (Национальный стандарт Российской Федерации).
5. Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення (ISO/IEC 12207:2008, IDT) : ДСТУ ISO/IEC 12207:2014. - [Чинний від 2016-01-01]. - К. : Держспоживстандарт України, 2016. - 106 с. - (Національний стандарт України).
6. Методи і засоби забезпечення безпеки. Критерії оцінки безпеки інформаційних технологій. Частина 3. Вимоги довіри до безпеки (ISO/IEC 15408-3:2008) : ДСТУ ISO/IEC 15408-3:2013. - [Чинний від 2013-12-05]. - К. : Держспоживстандарт України, 2013. - 142 с. - (Національний стандарт України).
7. Методический аппарат анализа и синтеза комплекса мер разработки безопасного программного обеспечения [Электронный ресурс] / А.В. Барабанов, А.С. Марков, В.Л. Цирлов : Международный научно-практический журнал «Программные продукты и системы» : Главный редактор С.В.Емельянов - 2015. -№4 (112). - С. 166-174. - Режим доступа к журн. <http://www.swsys.ru/index.php?page=article&id=4086&lang=>
8. Про скасування міждержавних стандартів в Україні, що розроблені до 1992 року : наказ ДП «УкрНДНЦ» №184 від 14.12.2015 р. [Електронний ресурс] - Режим доступа: http://ukrndnc.org.ua/files/2015/12_14/N184_2015-12-14.rtf
9. Системи управління якістю. Настанови щодо керування конфігурацією (ISO 10007:2003, IDT) : ДСТУ ISO 10007:2005. - [Чинний від 2007-08-01]. - К. : Науково-дослідний інститут метрології вимірювальних і управляючих систем (ДП «НДІ «Система»), 2007. - 12 с. - (Державний Стандарт України).
10. Технології програмування і захист інформаційних систем [Електронний ресурс] / О.Є. Коваленко, В.М. Антоненко // Інститут проблем математичних машин і систем НАН України (Академія ДПС України). - 2003. - Режим доступа: http://nc.asta.edu.ua/Kyrsi%202003/tezi/images_tezi/197.htm

УДК 004.4(05)

Бортник К.Я., Зух О.М., Лобода Р.В.

Луцький національний технічний університет

СОЦІАЛЬНА АВТОРИЗАЦІЯ

Бортник К.Я., Зух О.М., Лобода Р.В. Соціальна авторизація. Використання соціальної авторизації на своєму веб-сайті. Використовуючи ресурси найпопулярніших соціальних систем, та їхні програмні інтерфейси, можна добитися надійної авторизації на своєму сайті, яка буде комфортною для користувачів. Огляд бібліотеки Socialite.

Ключові слова: OAuth 2.0 - протокол авторизації, соціальна авторизація, бібліотека Socialite.

Bortnyk K.Y., Zukh O.N., Loboda R.V., Social authorization. Using social authorization on its website. Using the resources of the most popular social systems and their software interfaces, can achieve secure authentication on its website, which will be comfortable for users. Browse the library Socialite.

Keywords: OAuth 2.0 - authentication protocol, social authorization, library Socialite.

Бортник К.Я., Зух О.Н., Лобода Р.В. Социальная авторизация. Использование социальной авторизации на своем сайте. Используя ресурсы самых популярных социальных систем, и их программные интерфейсы, можно добиться надежной авторизации на своем сайте, которая будет комфортной для пользователей. Обзор библиотеки Socialite.

Ключевые слова: OAuth 2.0 - протокол авторизации, социальная авторизация, библиотека Socialite.

Для чого потрібна соціальна авторизація? За даними дослідницької компанії WebHostingBuzz, обов'язкова реєстрація не тільки відштовхує користувачів від подальшого користування сайтами, але і дає їм відчуття роздратування та втоми, яка виникає через необхідність пам'ятати велику кількість логінів і паролів від різних інтернет-ресурсів та сайтів.

Це не дивно, адже, як відомо, середній користувач Інтернет використовує від семи до тридцяти різних комбінацій логін / пароль, і необхідність створити та запам'ятати ще одну таку комбінацію його зовсім не радує.

Результати опитування дослідницької компанії Blue Research, проведені в 2012 році показали, що 54% людей схильні покинути сайт з обов'язковою реєстрацією і знайти інший ресурс, який не примушує їх створювати новий обліковий запис для входу в систему.

Використання соціальної авторизації на своєму веб-сайті дозволить їх власникам вирішити відразу всі описані вище проблеми. Реєстрація все ж таки відбуватиметься, але цей процес займатиме щитані секунди, не викликаючи у користувачів дискомфорту, і не заставляючи їх шукати більш комфортніший сайт.

Переваги соціальної авторизації перед звичайною

Кожен сайт, який претендує називатись сучасним, повинен надавати можливість користувачам авторизуватися, використовуючи одну або декілька найпопулярніших соціальних мереж.

Такий спосіб містить чимало переваг перед звичайним:

- Користувачам не потрібно проходити повноцінну реєстрацію, придумувати логін і пароль, набирати капчу, підтверджувати емейл, завантажувати аватар. В більшості випадків достатнього зробити 2 кліки і Ви вже зареєстровані на сайті.

- Дані, які нам приходять від соціальної мережі вже перевірені і підтверджені. Наприклад, якщо соціальна мережа ВК нам повертає e-mail користувача, то ми можемо бути впевнені, що він достовірний та підтверджений. Те ж саме стосується номера телефону та інших особистих даних користувача.

- Багато сервісів та соціальних мереж надають можливість увімкнути дворівневу авторизацію для зареєстрованого у них облікового запису. Тобто кожного разу (або раз в 30 днів) при вході в обліковий запис, на телефон приходить СМС з додатковим кодом, що значно підвищує безпеку облікових записів, а значить і безпеку для користувачів нашого сайту.

Проблеми, які можуть виникнути при соціальній авторизації, та шляхи їх вирішення

Як же повинна відбуватися авторизація на нашому сайті, враховуючи той факт, що соціальних мереж існує безліч? Що робити тоді, коли один і той самий користувач щоразу авторизується на нашому сайті, використовуючи іншу соціальну мережу?

До прикладу, користувач перший раз авторизувався через соціальну мережу Facebook, і здійснив певні покупки в інтернет – магазині. Наступного разу він знову авторизується в цьому магазині, але вже використовуючи іншу соціальну мережу - Vk.

По замовчуванню, повинен буде створитися зовсім інший, чистий профіль користувача із подібними особистими даними. Проте, така поведінка сайту призведе до чималих незручностей як зі сторони користувача, так і зі сторони адміністрації сайту.

Так, наприклад, користувач не зможе побачити своїх попередніх покупок (якщо це інтернет - магазин), не зможе побачити історію своєї попередньої активності, вподобань, тощо... В свою чергу, адміністрація сайту не зможе вести правдиву статистику, не зможе, наприклад, відслідкувати скільки покупок зробив користувач, та запропонувати на основі них якусь акцію або спеціальну пропозицію.

Отже, аби наш сайт був сучасним та зручним, перед нами постає наступна проблема: пов'язати кілька соціальних мереж з одним профілем. Причому все це повинне відбуватися неявно, і користувач не повинен нічого про це знати.

Для цього, нам потрібно відділити користувача, як окрему сутність, яка буде створюватися при першій авторизації на сайті, використовуючи любую із соціальних мереж.

Під час створення користувача, із першої соціальної мережі, через яку відбувся вхід, будуть отримані всі необхідні дані для сайту: ім'я, прізвище, аватар, та інші особисті дані. Далі при інших спробах авторизації, до створеного профіля користувача будуть приєднуватися інші соціальні мережі, як додаткова інформація, або опції. Сутність користувача при цьому не міняється, а залишається такою, як і була після першої авторизації на сайті. Точкою зв'язку між профілем користувача та соціальними мережами можуть виступати будь-які унікальні дані, такі як e-mail, телефон, і т.д.

Окрім того, що соціальні мережі можуть прив'язуватися до профілю користувача неявно (без його відому), більшість сучасних сайтів через свій інтерфейс надають можливість користувачам власноруч прикріплювати та відкріплювати аккаунти соціальних мереж від свого профілю.

Отже, використовуючи ресурси найпопулярніших соціальних систем, та їхні програмні інтерфейси, можна добитися надійної авторизації на своєму сайті, яка буде комфортною для користувачів, не буде для них складною, та забирати стільки багато часу.

Авторизація на базі протоколу OAuth2

Майже всі соціальні мережі, які дбають про безпеку користувачів та конфіденційність їхніх даних, пропонуть авторизацію по протоколу OAuth2.

Що ж таке OAuth2, та як він працює? OAuth 2.0 - протокол авторизації, що дозволяє видати одному сервісу права на доступ до ресурсів користувача на іншому сервісі. Протокол позбавляє від необхідності довіряти додатку логін і пароль, а також дозволяє видавати обмежений набір прав, а не все відразу.

OAuth 2.0 заснована на використанні базових веб-технологій: HTTP-запити, редіректи і т. п. Тому використання OAuth можливе в будь-якій платформі з доступом до інтернету і браузером: на сайтах, в мобільних і настільних додатках, і т.д.

Загальна схема роботи програми, що використовує OAuth, така:

- отримання авторизації
- звернення до захищених ресурсів

Результатом авторизації є ключ (зазвичай просто набір символів), використання якого є пропуском до захищених ресурсів. Звернення до них в найпростішому випадку відбувається по протоколу HTTPS із зазначенням в заголовках або в якості одного з параметрів отриманого ключа доступу.

Як використати соціальну авторизацію на своєму сайті. Огляд бібліотеки Socialite.

Розглянемо приклади реалізації соціальної авторизації по протоколу OAuth2 на сайті, який було розроблено з використанням фреймворка Laravel.

На сьогоднішній день існує безліч готових рішень та бібліотек для фреймворка Laravel, які допоможуть реалізувати такий спосіб авторизації не витрачаючи на розробку багато часу, та значно полегшуючи його підтримку в майбутньому.

Отож, перш ніж приступити до вирішення поставленої задачі, варто підшукати готову бібліотеку, в якій вже буде реалізована більша частина нашого функціоналу та основна логіка

авторизації. Шукати будемо в офіційному репозиторії пакетів для фреймворка Laravel за адресою <https://packagist.org/>

Найвідомішою та найбільш популярною такою бібліотекою є laravel/socialite, кількість установок якої налічує більше ніж півтора мільйони.

Визначною особливістю даної бібліотеки є те, що вона дозволяє працювати з багатьма соціальними мережами через однакові класи та об'єкти, як із одною соціальною мережею.

Для того, щоб установити бібліотеку на свій сайт, достатньою у консольному вікні в папці із своїм проектом ввести наступну команду:

```
composer require laravel/socialite
```

Після встановлення Socialite бібліотеки, потрібно зареєструвати новий провайдер у конфігураційному файлі проекту config/app.php, як це зроблено далі:

```
config/app.php
'providers' => [
    // Інші провайдери...
    Laravel\Socialite\SocialiteServiceProvider::class,
],
```

Потім потрібно додати аліас Socialite до масиву aliases у нашому app файлі конфігурацій:

```
config/app.php
'aliases' => [
    // Інші aliases...
    'Socialite' => Laravel\Socialite\Facades\Socialite::class,
],
```

На даному кроці установка бібліотеки повністю завершена. Тепер для нас стали доступними класи та об'єкти, які можна використовувати для написання своєї логіки, яка буде значно простішою та легшою для розуміння і підтримки іншими розробниками.

По замовчуванню дана бібліотека підтримує авторизацію через такі найпопулярніші соціальні мережі як Facebook, Twitter, Google, LinkedIn, GitHub та Bitbucket.

Та що ж робити, якщо ми хочемо авторизуватися через соціальну мережу, якої немає в переліку вище? Наприклад, соціальної мережі Vk.com, яка має неабияку популярність на теренах нашої держави, і якої користуються мільйони користувачів там теж немає.

Насправді, розробники бібліотеки Socialite про це вже подбали, і розробили можливість використання сторонніх плагінів, які дозволять “влити” логіку авторизації будь-якої іншої соціальної мережі в нашу бібліотеку.

Повний список плагінів, або так званих “соціальних провайдерів” можна віднайти на сайті <https://socialiteproviders.github.io/>. Даний список налічує більше ста різних соціальних мереж, починаючи від найвідоміших, і закінчуючи маловідомими, які не користуються особливою популярністю, проте мають таке ж саме право на життя і можуть використовуватися для авторизації. Саме в такому списку ми можемо віднайти соціальну мережу, як Vk.com, яку ми безумовно використаємо на своєму сайті, якщо хочемо зробити його зручним для користувачів.

Та все ж, якщо в даному списку не виявиться плагіна для потрібної нам соціальної мережі, ми можемо розробити її самотужки. Зробити це не важко, використовуючи ядро бібліотеки, і наслідуючи абстрактні класи та об'єкти.

Для того, щоб підключити такий плагін до основної бібліотеки, достатньо перейти до файла, що знаходиться за шляхом app/Providers/EventServiceProvider у папці із нашим проектом, і зареєструвати провайдер нашого плагіна:

```
/**
 * The event handler mappings for the application.
 *
 * @var array
 */
protected $listen = [
    \SocialiteProviders\Manager\SocialiteWasCalled::class => [
        // add your listeners (aka providers) here
        'SocialiteProviders\VKontakte\VKontakteExtendSocialite@handle',
    ],
];
```

Таким чином, засобами фреймворка Laravel та використовуючи функціонал бібліотеки Socialite, ми можемо розробити соціальну авторизацію будь якої складності, не витрачаючи на це велику кількість часу, та не розбираючись в нюансах та не перечитуючи документацію кожної із потрібних соціалок.

<https://laravel.com/>

<https://laravel.ru/>

<https://github.com/laravel/socialite>

<https://habrahabr.ru/company/mailru/blog/115163/>

<https://uk.wikipedia.org/wiki/OAuth>

<http://lpgenerator.ru/blog/2014/07/04/avtorizaciya-cherez-socialnye-seti-9-faktov-kotorye-dolzhen-znat-kazhdyy-marketolog/>

УДК 004.75; 641.8

Васильєва Л.В., доцент, Синюк Б.В., бакалавр
Харківський національний університет імені В.Н. Каразіна

ДОСЛІДЖЕННЯ СТУПЕНЯ БЕЗПЕКИ ВИСОКОТЕХНОЛОГІЧНИХ 3D-ПРИНТЕРІВ

Васильєва Л.В., Синюк Б.В. Дослідження ступеня безпеки високотехнологічних 3D-принтерів. У статті висвітлені загальні питання безпечного функціонування 3D-принтерів, та запобігання негативного впливу можливих ризиків при використанні цієї комп'ютерної технології. Також у статті приводиться дослідження ризиків при 3D-друці різними 3D-принтерами та матеріалами.

Ключові слова: 3D-друк, 3D-принтери, безпека, УДЧ, технології 3D-друку.

Васильєва Л.В., Синюк Б.В. Исследование степени безопасности высокотехнологичных 3D-принтеров. В статье освещены общие вопросы безопасного функционирования 3D-принтеров и предотвращения негативного влияния возможных рисков при использовании этой компьютерной технологии. Также в статье приводится исследование рисков при 3D-печати различными 3D-принтерами и материалами.

Ключевые слова: 3D-печать, 3D-принтеры, безопасность, УДЧ, технологии 3D-печати.

Vasilyeva L.V., Sinyuk B.V. Investigation of the security degree of high-tech 3D printers. The article covers general issues of safe operation of 3D printers and prevention of the negative impact of risk's possible when using this computer technology. Also in the article is a study of the risks of 3D printing with using various 3D printers and materials.

Keywords: 3D-printing, 3D-printers, security, UDC, 3D-printing technologies.

Вступ. Світ не стоїть на місці, зростає прогрес зростають і ризики, в разі друку на високотехнологічних пристроях - 3D-принтери. Для забезпечення високої якості друку і застосування майже в усіх сферах виробництва, а також для різноманітності виготовляємих об'єктів, частіше всього використовуються різні методи 3D-друку. Як з безпечними речовинами, так і небезпечними, а часом і токсичними. Знання формальних моделей поведінки і методів безпеки для роботи з 3D-принтерами і також при 3D-друці, забезпечує високу безпеку фахівця.

Самі принтери покликані допомагати людям, але неохайне поводження може призвести до трагічних випадків, і навіть летальних наслідків. Головне правило при 3D-друці - це завжди контролювати процес виробництва нової моделі.

3D-друк - це процес створення фізичного об'єкта методом пошарового накладання матеріалу за його комп'ютерною 3D-моделлю. Друк об'ємних фігур в сучасному суспільстві починають застосовувати безліч галузей виробництва, так як вона дозволяє створювати унікальні об'єкти, як за матеріалами, так і по точності моделі. Звичайно ж ні один процес 3D-друку не залишається без пристрою, який його виконує, а саме 3D-принтера [3].

Постановка наукової проблеми. Метою даної роботи є аналіз специфікації та безпеки роботи з різноманітними типами 3D-принтерами з метою запобігання нещасних випадків на виробництві та побуті.

Аналіз існуючих способів.

На ринку зараз є різноманітні 3D-принтери: 3D-принтер конструктор, не комерційні, професійні (медичні, будівельні, харчові, текстильні, біологічні тощо), виробничі і т.д. Всі вони мають свої переваги та недоліки, проте кожен з них також несе в собі загрозу, і є небезпечними, якщо нехтувати правилами ОП. Тому існують стандартні правила безпеки, майже, з будь-яким 3D-принтером.

Проаналізувавши, які існують у сфері сучасних концепцій ОП нормативи та інструкції по роботі з 3D-принтерами, можна виділити основні пункти з безпеки:

- При роботі на 3D-принтері не допускається розташування робочого місця в приміщеннях без наявності природної або штучної вентиляції.
- Перед початком роботи на 3D-принтері перевірити його справність (спочатку візуально, а потім, провівши кілька маленьких тестів) і перевірити захисне заземлення обладнання.
- Переконавшись, що мікроклімат в приміщенні дозволяє працювати з цим обладнанням.
- Не підпускати до 3D-принтера осіб, які не знайомі з правилами безпечної поведінки з цим обладнанням

- Вмикати і вимикати 3D-принтер слідкує тільки вимикачем. Категорично забороняється витягувати вилку шнура 3D-принтера з розетки під час роботи.
- Забороняється торкатись робочих частин 3D-принтера (екструдер, робочий стіл для друку і т.д.). Також знімати або деактивувати захисні компоненти 3D-принтера.
- Забороняється залишати увімкнений 3D-принтер без нагляду.
- Місце, де стоїть 3D-принтер має повністю відповідати нормам протипожежної безпеки.
- Розбирати і ремонтувати 3D-принтер самому забороняється, ці роботи може виконувати тільки фахівець.
- Загальний час безпосередньої роботи з 3D-принтером протягом робочого дня не повинен перевищувати 5-6 годин.
- Після роботи потрібно ретельно провітрити приміщення з 3D-принтером [4].

Виходячи з особливостей технології 3D-друку специфічної для різних типів 3D-принтерів можна виділити основні ризики:

I. Група 3D-принтерів - це ті, які щось виштовхують, виливають, або розпилюють:

1) *FDM (fused deposition modeling)* - принтери які видавлюють якийсь матеріал шар за шаром через сопло-дозатор.

2) *Технологія Polyjet* - фотополімер маленькими дозами вистрілюється з тонких сопел, як при струменевому друці, і відразу полімеризується на поверхні виготовляемого виробу під впливом УФ випромінювання. PolyJet друкує різними матеріалами.

Застосування: промислове прототипування і медицина.

3) *LENS (LASER ENGINEERED NET SHAPING)* - матеріал у формі порошку видувається з сопла і потрапляє на сфокусований промінь лазера. Частина порошку пролітає повз, а та частина, яка потрапляє у фокус лазера миттєво спікається і шар за шаром формує тривимірну деталь. Саме за такою технологією друкують сталеві і титанові об'єкти.

Застосування: наприклад, титанові лопатки для турбін з внутрішніми каналами охолодження.

4) *LOM (laminated object manufacturing)* - тонкі ламіновані листи матеріалу вирізаються за допомогою ножа або лазера і потім спікається або склеюються в тривимірний об'єкт. Таким чином друкують 3D-моделі з паперу, пластику або з алюмінію.

У даній групі 3D-принтерів головною небезпекою є токсичність деяких використовуваних матеріалів. Також можливе засмічення сопел тим чи іншим матеріалом, що призводить до займання (наприклад 3D-принтер, який друкує пластмасою).

II. Група 3D-принтерів - це ті, які щось спікають або склеюють

1) *SL (Stereolithography) стереолітографія* - є невелика ванна з рідким полімером. Промінь лазера проходить по поверхні, і в цьому місці полімер під впливом УФ полімеризується. Платформа з деталлю піднімається вгору, лазер відповідно розташований знизу.

2) *LS (laser sintering) лазерне спікання* - схоже на SL, тільки замість рідкого фотополімеру використовується порошок, який спікається лазером.

3) *3DP (three dimensional printing)* - на матеріал в порошкової формі наноситься клей, який пов'язує гранули, потім поверх проклеєного шару наноситься свіжий шар порошку, і так далі. На виході, виходить матеріал sandstone (за властивостями схожий на гіпс).

У даній групі 3D-принтерів головними небезпеками є: у більшості випадків фотополімер буває токсичним, тому при роботі з ним треба користуватися засобами захисту і респіраторами; деякі порошки для спікання бувають вибухонебезпечними, вони повинні зберігатися в камерах з азотом, що в свою чергу вимагає підвищених вимог ОП; при даних видах 3D-друку використовується дуже висока температура [2].

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

В Іллінойському технологічному інституті групою вчених було проведено дослідження, яке показало, що некомерційні 3D-принтери мають особливість виділяти у повітря небезпечні для нашого організму частинки нанометрових розмірів (ультрадисперсні частинки - УДЧ).

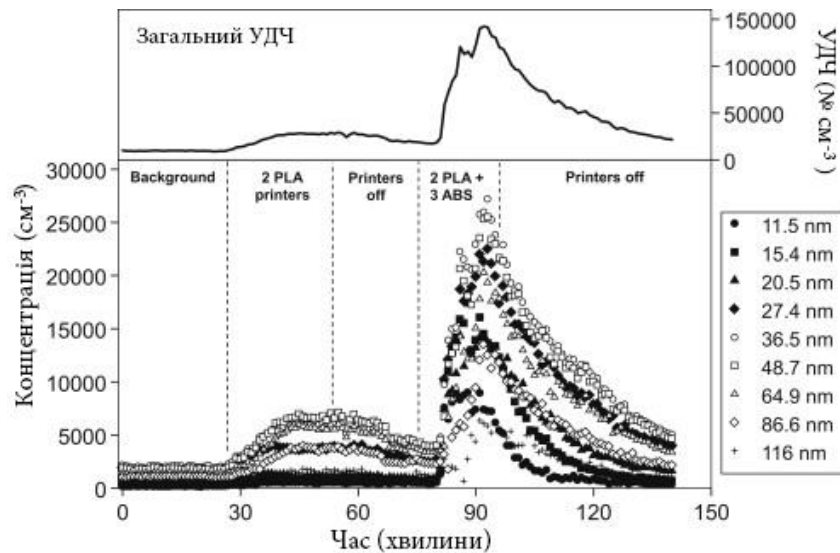


Рис. 1. (<100 nm) Зміна концентрації УДЧ з дозволеної, і під час роботи 3D-принтерів.

На рис. 1 зображена зміна концентрації УДЧ з дозволеної, тієї, що спостерігається у фоновому режимі (3D-принтери вимкнені та до цього часу не вмикалися в даному приміщенні) до початку роботи 3D-принтерів, а саме – спочатку працювали два 3D-принтери, що друкували PLA пластмасою (виробляється з біоматеріалу), потім спостерігаємо період їх простою і наступний період, де вже працює 2 3D-принтери з PLA та 3 3D-принтери з ABS пластмасою. Останній період простою 3D-принтерів.

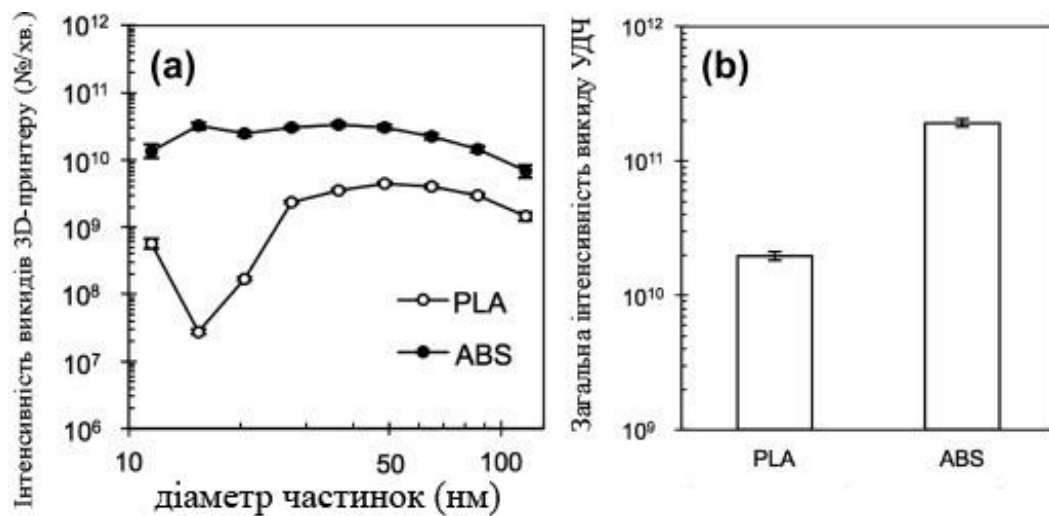


Рис. 2. Окремі показники викидів УДЧ від 3D-принтерів, що використовують два типи термопластичної вихідної сировини в даному дослідженні: (а) коефіцієнти викидів виміряних розмірів (11.5-116 nm) і (б) загальні коефіцієнти викидів УДЧ (<100 nm).

У табл. 1 описані одні і ті ж центральні оцінки коефіцієнтів викидів УДЧ з рис. 2 поряд з діапазоном невизначеності, розраховані для кожного розміру [1].

Таблиця 1. Оцінки інтенсивності викидів від окремих 3D принтерів, що використовують різні термопластичні вихідні матеріали.

Розмір частинок	Інтенсивність викидів УДЧ на принтер (№ хвил. ⁻¹)			
	2 PLA (Період 2)		2 PLA та 3 ABS (Період 3)	
	Центральна оцінка	Діапазон	Центральна оцінка	Діапазон
11,5 нм	$5,8 \times 10^8$	$[4,7-6,8] \times 10^8$	$1,4 \times 10^{10}$	$[1,1-1,7] \times 10^{10}$
15,4 нм	$2,8 \times 10^7$	$[2,6-3,0] \times 10^7$	$3,2 \times 10^{10}$	$[2,8-3,6] \times 10^{10}$
20,5 нм	$1,7 \times 10^8$	$[1,6-1,8] \times 10^8$	$2,5 \times 10^{10}$	$[2,2-2,7] \times 10^{10}$
27,4 нм	$2,4 \times 10^9$	$[2,2-2,5] \times 10^9$	$3,0 \times 10^{10}$	$[2,8-3,3] \times 10^{10}$
36,5 нм	$3,6 \times 10^9$	$[3,4-3,7] \times 10^9$	$3,4 \times 10^{10}$	$[3,2-3,7] \times 10^{10}$
48,7 нм	$4,5 \times 10^9$	$[4,2-4,7] \times 10^9$	$3,1 \times 10^{10}$	$[2,8-3,3] \times 10^{10}$
64,9 нм	$4,0 \times 10^9$	$[3,8-4,3] \times 10^9$	$2,2 \times 10^{10}$	$[2,0-2,4] \times 10^{10}$
86,6 нм	$3,0 \times 10^9$	$[2,8-3,2] \times 10^9$	$1,5 \times 10^{10}$	$[1,3-1,6] \times 10^{10}$
116 нм	$1,5 \times 10^9$	$[1,2-1,7] \times 10^9$	$6,9 \times 10^9$	$[5,4-8,5] \times 10^9$
Разом УФПС (<100 нм)	$2,0 \times 10^{10}$	$[1,9-2,0] \times 10^{10}$	$1,9 \times 10^{11}$	$[1,8-2,0] \times 10^{11}$

Незважаючи на це, настільні 3D-принтери, досліджені тут можуть бути класифіковані як «високі випромінювачі» з УДЧ рівнів викидів, більш ніж 1010 частинок на хвилину. Також слід зазначити, що викиди УДЧ на порядок вище у 3D-принтерів, що використовують ABS пластмасу, аніж PLA.

Висновки

Щорічно збільшується кількість підприємств зі шкідливими та небезпечними умовами праці. Недооцінка, ігнорування факторів створення відповідних умов для безпечної праці завдають непоправної шкоди здоров'ю працівників, негативно впливає на рівень продуктивності праці. І галузь 3D-індустрії також є областю праці, яка схильна до роботи зі шкідливими і навіть токсичними речовинами та матеріалами, проте вона відрізняється своїми особливими заходами безпеки та засобами, призначеними для роботи з небезпечними матеріалами.

Це дослідження говорить, що слід обережно використовувати деякі типи і підтипи 3D-принтерів, а також з підвищеною обережністю працювати в приміщеннях, де забезпечені не всі вимоги для безпечної роботи з 3D-принтерами, особливо с настільними (некомерційними) 3D-принтерами. Дотримуватися основних правил безпеки на підприємствах та правил безпеки роботи зі шкідливими речовинами.

Перспективи подальших досліджень.

3D-друк відносно ненова річ, проте до сих пір не повністю вивчена. З кожним роком з'являються нові методи та типи 3D-друку. Сама промисловість лише починає переходити на роботу с 3D-принтерами, що виробляють різну продукцію. А отже вплив різних методів та типів 3D-друку на здоров'я людини лише починає вивчатись. Знадобиться ще як мінімум 5, а то і десять років, перш ніж появляться дослідження впливу на здоров'я людини, бо сама спеціальність роботи с 3D-принтерами в промислових масштабах з'явилася декілька років назад.

1. Brent Stephens, Parham Azimi, Zeineb El Orch, Tiffanie Ramous "Ultrafine particle emissions from desktop 3D printers" - *Atmospheric Environment*, Volume 79, November 2013, Pages 334–339 [Електронний ресурс]. – Режим доступу: <http://www.sciencedirect.com/science/article/pii/S1352231013005086>
2. Классификация 3D принтеров (7 технологий 3D печати) [Електронний ресурс]. – Режим доступу: <https://geektimes.ru/post/208906/>
3. 3D печать, коротко и максимально ясно [Електронний ресурс]. – Режим доступу: http://media.wix.com/ugd/e25109_f92ab8c4d9034c13b1909a293439baf6.pdf
4. Инструкция по охране труда при работе с 3D-принтером [Електронний ресурс]. – Режим доступу: <http://3dtoday.ru/blogs/kirillll/instruction-on-labor-protection-when-working-with-the-3d-printer/>
5. Охрана праці в Україні : Нормативні документи/ Упоряд. О. М. Роїна, Ред. О. А. Кривенко. -2-ге вид., виправлене і доповнене. -К.: КНТ, 2006. -418 с.
6. Лапін В. Основы охраны праці : Навчальний посібник/ Віктор Лапін,; М-во освіти і науки України, НБУ, ЛБІ. -Львів: ЛБІ НБУ, 2004. -142 с.

7. Охорона праці: європейські і міжнародні стандарти та законодавство України (порівняльний аналіз) : Науково-практичний посібник: У 2-т./ Упор.: В. С. Венедіктов, В. П. Грохольський, М. І. Іншин та ін.; За ред. В. С. Венедіктова; М-во юстиції України, Державний департамент з питань адаптації законодавства, Українська асоціація фахівців трудового права. -Харків-Київ. -2006. - Т. 1. -2006. -713 с.. -Предм. покажч.: с. 713
8. Ярошевська В. М. Охорона праці в галузі : Навчальний посібник/ В. М. Ярошевська, В. Й. Чабан; М-во науки і освіти України, Український держ. ун-т водного господарства та природокористування. -Київ: ВД "Професіонал, 2004. -286 с.
9. Гандзюк М. Основи охорони праці : Підручник для студ. вуз./ Ми-хайло Гандзюк, Євген Желібо, Модест Халімовський, За ред. Михайла Гандзюка. -2-е вид.. -К.: Каравела, 2004. -405 с.
10. Геврик Є. Охорона праці : Навчальний посібник/ Є.О. Геврик,. -К.: Ельга: Ніка-Центр, 2003. – 279 с.

УДК 514.18

Верещага В.М., д.т.н.¹, Адоньєв Є.О., к.т.н.²

¹ - Мелітопольський державний педагогічний університет ім. Б. Хмельницького

² - Запорізький національний університет

КОМПОЗИЦІЙНИЙ МЕТОД УТВОРЕННЯ Б-ПОВЕРХОНЬ

Верещага В.М., Адоньєв Є.О. Композиційний метод утворення Б-поверхонь. Введено поняття «Б-поверхня» (Балюби поверхня). Показані принципи композиційного методу утворення Б-поверхонь та доведено необхідність його подальшої розробки. Сформульовано ознаки та правила створення точкових рівнянь, які описують Б-поверхні. Показано властивості, що їх визначають, та можливості щодо перетворень Б-поверхонь, вказано напрями їхнього застосування.

Ключові слова: геометричне моделювання, точкове рівняння, параболічна поверхня, композиція, суперпозиція.

Верещага В.М., Адоньєв Е.А. Композиционный метод образования Б-поверхностей. Введено понятие «Б-поверхность» (Балюбы поверхность). Показаны принципы композиционного метода образования Б-поверхностей и доказана необходимость его дальнейшей разработки. Сформулированы признаки и правила построения точечных уравнений, которые описывают Б-поверхности. Показаны свойства, которые их определяют и возможности преобразования Б-поверхностей, указаны направления их применения.

Ключевые слова: геометрическое моделирование, точечное уравнение, параболическая поверхность, композиция, суперпозиция.

Vereschaha V.M., Adoniev Y.O. Compositional method of B-surfaces formation. The notion of "B-surface" (Balyuby surface). The following principles of compositional method of B-surfaces formation and the necessity of its further development. Formulated signs and rules for creating point equations that describe the B-surface. Showing properties that define them, and the possibilities for transformation of B-surfaces specified areas of their application.

Keywords: geometric modeling, point equation, parabolic surface, composition, superposition.

Постановка проблеми. Традиційні геометро-математичні методи утворення поверхонь, моделювання процесів та дослідження явищ і станів виконується з використанням способів комбінаційного характеру, у яких складові - математичні елементи є взаємно залежними, при цьому, застосовуються різні техніки для знаходження кореляцій між складовими [10]. Наявність взаємної залежності між елементами геометро-математичної моделі завжди висуває до неї певні обмеження за кількістю факторів, за розмірами матриць, тощо, а це призводить до створення неадекватних моделей, розрахунки за якими, здебільшого, будуть мати похибки, в результаті чого будуть прийматися помилкові рішення. Для підвищення адекватності моделі розробляються різні способи виявлення головних факторів, елементів, компонент, складових геометричного характеру, тощо, що надає певні покращення моделі, однак, при цьому, не знімається обмеження з кількості вихідних факторів, що характеризують процес [6, 7]. Тому актуальною є проблема розробки методу моделювання, який мав би можливість, при створенні моделі, включати максимально необхідну кількість факторів досліджуваного процесу. На нашу думку, розв'язок проблеми збільшення вихідної інформації у моделюванні криється у розробці композиційних моделей, у яких не існує взаємно обумовленого зв'язку між її елементами.

Огляд літературних джерел. Ідея розробки композиційних методів моделювання поверхонь виникла після досконалого пізнання та осмислення точкового числення Балюби-Найдиша [1]. Були опрацьовані принципи побудови у точковому численні поверхонь типу "Лупа" [8, 9], які не сформовані у вигляді повноцінного методу моделювання, але задають певні напрямки досліджень і використовуються у даній роботі. Тому точкове БН-числення, в тому числі, методика побудови параболічних поверхонь [2, 4, 5] стало підґрунтям для виникнення композиційного методу моделювання, який запропоновано авторами і уперше розглядається у цій статті.

Ціль та завдання статті. На прикладі Б-поверхні розглянути основи та необхідність розробки композиційного методу геометричного моделювання, показати його переваги та означити перспективи застосування.

Основна частина. Розглянемо параболічну поверхню Балюби (БПП), що задана точками $x_{11}, \dots, x_{33}$, тобто x_{ij} , для $1 \leq i, j \leq 3$ (рис.1).

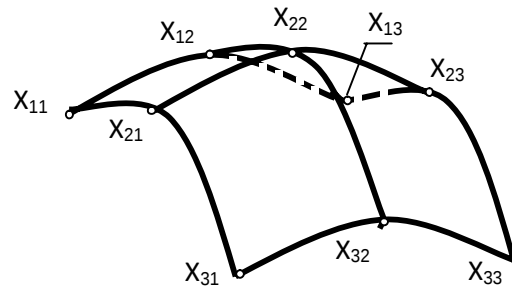


Рис. 1. Схема БПП у просторі

Нехай у повздовжньому напрямі визначені параболи M_1 , M_2 та M_3 з однаковими параметрами

$$p_1 = 4u^2 - 3u + 2; \quad q_1 = 5u^2 + 4; \quad (1)$$

$$M_1 = (x_{11} - x_{12})p_1 + (x_{13} - x_{12})q_1 + x_{12};$$

$$M_2 = (x_{21} - x_{22})p_1 + (x_{23} - x_{22})q_1 + x_{22}; \quad (2)$$

$$M_3 = (x_{31} - x_{32})p_1 + (x_{33} - x_{32})q_1 + x_{32};$$

а у поперечному напрямі параметри

$$p_2 = 2v^3 - v^2 - 1; \quad q_2 = v^2 - 4v + 2, \quad (3)$$

що є однаковими для усіх N_1 , N_2 , N_3 парабол:

$$N_1 = (x_{11} - x_{21})p_2 + (x_{31} - x_{21})q_2 + x_{21};$$

$$N_2 = (x_{12} - x_{22})p_2 + (x_{32} - x_{22})q_2 + x_{22}; \quad (4)$$

$$N_3 = (x_{13} - x_{23})p_2 + (x_{33} - x_{23})q_2 + x_{23}.$$

Враховуючи (1) та (3), можемо записати:

$$r_1 = -9u^2 + 3u - 5; \quad r_2 = -2v^3 + 4v. \quad (5)$$

З урахуванням (5) можемо (2) та (3) записати у наступному вигляді:

$$M_1 = x_{11}p_1 + x_{12}r_1 + x_{13}q_1;$$

$$M_2 = x_{21}p_1 + x_{22}r_1 + x_{23}q_1; \quad (6)$$

$$M_3 = x_{31}p_1 + x_{32}r_1 + x_{33}q_1;$$

$$N_1 = x_{11}p_2 + x_{21}r_2 + x_{31}q_2;$$

$$N_2 = x_{12}p_2 + x_{22}r_2 + x_{32}q_2; \quad (7)$$

$$N_3 = x_{13}p_2 + x_{23}r_2 + x_{33}q_2.$$

Складемо схему утворення елементів-функцій a_{ij} :

$$\begin{array}{ccc|c} p_1 & r_1 & q_1 & p_2 \\ p_1 & r_1 & q_1 & r_2 \\ p_1 & r_1 & q_1 & q_2 \end{array} \quad (8)$$

Тоді визначимо елементи-функції як добутки:

$$a_{11} = p_1p_2 = (4u^2 - 3u + 2)(2v^3 - v^2 - 1);$$

$$a_{12} = r_1p_2 = (-9u^2 + 3u - 5)(2v^3 - v^2 - 1);$$

$$a_{13} = q_1p_2 = (5u^2 + 4)(2v^3 - v^2 - 1);$$

$$a_{21} = p_1r_2 = (4u^2 - 3u + 2)(-2v^3 + 4v);$$

$$\begin{aligned}a_{22} &= r_1 r_2 = (-9u^2 + 3u - 5)(-2v^3 + 4v); \\ a_{23} &= q_1 r_2 = (5u^2 + 4)(-2v^3 + 4v); \\ a_{31} &= p_1 q_2 = (4u^2 - 3u + 2)(v^2 - 4v + 2); \\ a_{32} &= r_1 q_2 = (-9u^2 + 3u - 5)(v^2 - 4v + 2); \\ a_{33} &= q_1 q_2 = (5u^2 + 4)(v^2 - 4v + 2);\end{aligned}\quad (9)$$

Точкове рівняння параболічної поверхні Балюби (БПП) матиме вигляд

$$M = \sum_{i,j=1}^3 x_{ij} a_{ij} \quad (10)$$

Розрахуємо функції-параметри a_{ij} , результати розрахунків для значень $u, v = 0; 0.2; 0.4; 0.5; 0.6; 0.8; 1.0$, зведемо у таблицю 1:

Таблиця 1. Результати розрахунків функцій-параметрів

a_{ij}	u, v						
	0	0,2	0,4	0,5	0,6	0,8	1
a_{11}	-2	-1,59744	-1,48608	-1,5	-1,52192	-1,33056	0
a_{12}	5	4,87424	5,40768	5,75	5,97632	5,14976	0
a_{13}	-4	-4,3008	-4,9536	-5,25	-5,3824	-4,4352	0
a_{21}	0	1,22304	2,11968	2,625	3,22752	4,70016	6
a_{22}	0	-3,73184	-7,71328	-10,0625	-12,67392	-18,19136	-22
a_{23}	0	3,2928	7,0656	9,1875	11,4144	15,6672	18
a_{31}	4	1,9344	0,8064	0,375	-0,0656	-1,2096	-3
a_{32}	-10	-5,9024	-2,9344	-1,4375	0,2576	4,6816	11
a_{33}	8	5,208	2,688	1,3125	-0,232	-4,032	9
$\sum a_{ij}$	1	1	1	1	1	1	1

Покажемо графічно (рис. 2) результати розрахунків a_{ij} , що наведені у табл. 1.

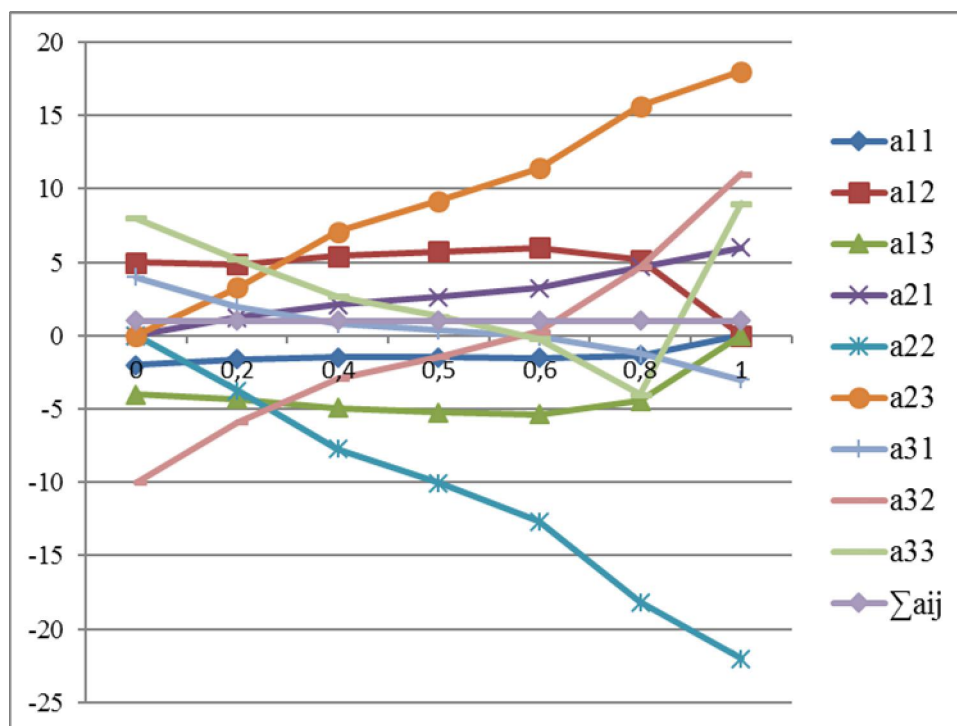


Рис. 2. Графіки функцій-параметрів a_{ij} .

Як бачимо, суперпозиція $\sum_{i,j=1}^3 a_{ij}$ для усіх значень параметрів u, v , які прийнято у табл. 1, дорівнює одиниці (останній рядок табл. 1).

Чому при будь-яких значеннях p_1, q_1, p_2, q_2 суперпозиція функцій-параметрів a_{ij} буде дорівнювати одиниці? Виходячи з теорії точкового БН-числення [1], точкові рівняння для усіх геометричних фігур мають область значення $0 \leq t \leq 1$. Тому, якщо параметри p_i, q_i можемо обирати довільно (аналогічно, як дві координати точки на площині), то третій параметр r_i необхідно обирати із умови $p_i + q_i + r_i = 1$. Якщо, врахувавши це та використавши схему утворення функцій-параметрів (елементів) a_{ij} (8), то можемо записати:

$$\begin{bmatrix} p_1 p_2 & r_1 p_2 & q_1 p_2 \\ p_1 r_2 & r_1 r_2 & q_1 r_2 \\ p_1 q_2 & r_1 q_2 & q_1 q_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{33} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (11)$$

Якщо розглядати окремо кожний рядок матриці (11), то бачимо, що кожен з них дорівнює добутку одиниці на відповідний параметр p_2, r_2, q_2 :

$$\begin{aligned} (p_1 + q_1 + r_1)p_2 &= 1 \cdot p_2; \\ (p_1 + q_1 + r_1)r_2 &= 1 \cdot r_2; \\ (p_1 + q_1 + r_1)q_2 &= 1 \cdot q_2. \end{aligned}$$

Враховуючи, що $p_2 + r_2 + q_2 = 1$, маємо: $1 \cdot p_2 + 1 \cdot r_2 + 1 \cdot q_2 = p_2 + r_2 + q_2 = 1$, а звідси випливає, що суперпозиція $\sum_{i,j=1}^3 a_{ij} = 1$.

Твердження 1. Суперпозиція функцій-параметрів a_{ij} у будь-якої параболічної поверхні Балюби (Б-поверхні) завжди дорівнює одиниці.

Наявність такої ознаки Б-поверхонь вказує на те, що параметри p_1, q_1 першого напрямку u на цій поверхні можуть бути обрані довільно, але вибір r_1 потрібно робити, дотримуючись певної функціональної залежності.

Для другого напрямку Б-поверхні – v також параметри p_2, q_2 обираються довільно, а параметр r_2 є взаємно зумовленим з параметрами p_2 та q_2 (тобто їхньою комбінацією).

Розглянемо другу властивість Б-поверхонь. Якщо визначити відношення першого рядка до другого матриці (11), то отримаємо матрицю-рядок:

$$\left\{ \frac{p_1 p_2}{p_1 r_2}, \frac{r_1 p_2}{r_1 r_2}, \frac{q_1 p_2}{q_1 r_2} \right\} = \{1 \quad 1 \quad 1\} \frac{p_2}{r_2}. \quad (12)$$

Отриманий результат з (12) свідчить про те, що елементи першого та другого рядків матриці (11) є пропорційними один до одного з коефіцієнтом пропорційності $\frac{p_2}{r_2}$.

Аналогічним чином покажемо пропорційність елементів другого та третього рядків матриці (11):

$$\left\{ \frac{p_1 r_2}{p_1 q_2}, \frac{r_1 r_2}{r_1 q_2}, \frac{q_1 r_2}{q_1 q_2} \right\} = \{1 \quad 1 \quad 1\} \frac{r_2}{q_2}. \quad (13)$$

Покажемо, що пропорційними є відношення між елементами першого та другого, другого та третього стовпчиків матриці (11):

$$\left\{ \begin{array}{c} \frac{p_1 p_2}{r_1 p_2} \\ \frac{p_1 r_2}{r_1 r_2} \\ \frac{p_1 q_2}{r_1 q_2} \\ \frac{p_1 q_2}{r_1 q_2} \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \end{array} \right\} \frac{p_1}{r_1}; \quad \left\{ \begin{array}{c} \frac{r_1 p_2}{q_1 p_2} \\ \frac{r_1 r_2}{q_1 r_2} \\ \frac{r_1 q_2}{q_1 q_2} \\ \frac{r_1 q_2}{q_1 q_2} \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \end{array} \right\} \frac{r_1}{q_1}; \quad (14)$$

Також пропорційними є перший та третій рядки та стовпчики. Як відомо [3], достатньо пропорційності лише між двома рядками або стовпчиками з (11), тоді визначник $\det A$ цієї матриці буде дорівнювати нулю:

$$\det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = 0.$$

Твердження 2. Визначник, складений із функцій-параметрів a_{ij} будь-якої Б-поверхні дорівнює нулю.

Зауважимо, що і перше, і друге твердження, кожне окремо, є необхідною, але недостатньою умовою для визначення Б-поверхонь. Отже, твердження 3 визначає і необхідну, і достатню умови.

Твердження 3. Якщо суперпозиція $\sum_{i,j=1}^3 a_{ij}$ елементів матриці $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$

дорівнює одиниці, тобто $\sum_{i,j=1}^3 a_{ij} = 1$, а визначник $\det A = 0$, то поверхня, що побудована на будь-яких вихідних даних $x_{11}, \dots, x_{33}$ є Б-поверхнею.

Кожна точка M_{ij} (10) Б-поверхні вираховується як сума частин a_{ij} від одиниці, що помножена на відповідну координату x_{ij} , тобто є композицією частин значень координат вихідних точок. Оскільки вихідні дев'ять дійсних точок є незалежними, то їх можна обрати таким чином, щоб Б-поверхня стала площиною, кривою лінією, прямою або точкою. Існуючі геометро-математичні методи надають поверхні, що мають комбінаційний характер утворення, тобто такі, що описані рівнянням або системою рівнянь, де зміна вихідних даних може привести до зміни самого рівняння.

І навпаки, Б-поверхні мають композиційний характер, у яких незалежні координати (точки), як елементи пазлу, складають картину на основі нескінченної кількості композицій. Б-поверхні, в залежності від того, дискретно чи неперервно визначені їхні параметри, є дискретними або неперервними.

На нашу думку, Б-поверхні є новим типом поверхонь композиційного характеру.

Висновки та перспективи подальших досліджень. Відкрито новий тип поверхонь композиційного характеру – Б-поверхні, вперше сформульовані їхні ознаки та правила створення точкових рівнянь, що їх описують. Показано властивості щодо виродження Б-поверхонь у площину, криву або пряму лінії і, навіть, у точку, що дало можливість розробити спосіб розгортання-згортання чарунок, на базі якого буде розроблено метод моделювання багатофакторних процесів.

1. Балюба И.Г., Найдыш В.М. Точечное исчисление / Под ред. В.М. Верещаги: Учебное пособие. – Мелитополь: Изд-во МГПУ им. Б. Хмельницького. 2015. – 234 с.
2. Балюба И.Г. Точечное уравнение параболической дуги кривой второго порядка / И.Г. Балюба, А.И. Бумага // Збірник тез доповідей і повідомлень VI Міжнародної наукової конференції молодих вчених, аспірантів, студентів. – Макіївка: ДонНАБА, 2007 – с. 64.
3. Бронштейн И.Н., Семендяев К.А. Справочник по математике для инженеров и учащихся ВТУЗов. – М., 1980, 976 с.

4. Бумага А.И. Теоретические основы конструирования геометрических объектов многомерного пространства в БН-исчислении / А.И. Бумага, А.В. Найдыш, Е.В. Конопацкий, О.А. Чернышова // Научные итоги: достижения, перспективы, гипотезы: Сб. докл. XVIII юбилейной междунар. научн.-практ. конф. (28 ноября 2013 г.) – Вып. 18. С. 151-154.
5. Бумага А.И. Точкове рівняння дуги параболи другого порядку. Геометрическое и компьютерное моделирование: энергосбережение, экология, дизайн: Матер. IX Крымской междунар. науч.-практ. конф. (24-28 сентября 2012 г.). Симферополь. Міжвідомчий науково-технічний збірник. Прикладна геометрія та інженерна графіка (спецвипуск). К.: КНУБА, 2012. – Вип. 90. – С. 49-52
6. Ильин В.А., Позняк Э.Г. Основы математического анализа. ч. 1-2, - М., 1971-73.
7. Колмогоров А.Н., Фомин С.В. Элементы теории функций и функционального анализа, 4 изд., М., 1976.
8. Кучеренко В.В. Реконструкція способом "Лупа" дискретно представленої поверхні земельної ділянки на основі рівномірної сітки у плані / В.В. Кучеренко, В.М. Верещага, І.Г. Балюба, Є.В. Конопацький // Прикладна геометрія та інженерна графіка. Праці / Таврійський державний агротехнологічний університет. – Вип. 4, т. 55. – Мелітополь: ТДАТУ, 2012 – С. 143-147.
9. Кучеренко В.В. Формалізовані геометричні моделі нерегулярної поверхні для гіперкількісної дискретної скінченної множини точок: дис. ... канд. техн. наук, 05.01.01 / Вадим Володимирович Кучеренко, Таврійський державний агротехнологічний університет. – Мелітополь, 2013. – 232 с.
10. Математическая энциклопедия / Гл. ред. И.М. Виноградов. – М.: Советская энциклопедия. тт. 2, 3, 4, 5. 1979 – 1984 гг.

УДК 621. 941

Т.Н. Гальчук

Луцький національний технічний університет

КОМП'ЮТЕРНА ОБРОБКА СТАТИЧНИХ ДАНИХ КОНТРОЛЮ ТОЧНОСТІ ВИРОБІВ МАШИНОБУДУВАННЯ

Гальчук Т.Н. Комп'ютерна обробка статичних даних контролю точності виробів машинобудування.

Показана ефективність комп'ютерної обробки статистичних даних контролю точності виробів машинобудування. Представлено реалізацію на виробництві контролю точності якості продукції за допомогою обчислювальної техніки для обробки результатів вимірювань.

Ключові слова: контроль, точність, виріб, статистичний аналіз, програма.

Рис. 6. Літ. 6.

Гальчук Т.Н. Компьютерная обработка статистических данных контроля точности изделий машиностроения
Показана эффективность компьютерной обработки статистических данных контроля точности изделий машиностроения. Представлено реализацию на производстве контроля точности по качеству продукции с использованием вычислительной техники для обработки результатов измерений.

Ключевые слова: контроль, точность, изделие, статистический анализ, программа.

T.N. Gal'chuk. Computer processing of static information accuracy control products of mechanical engineering.

The efficiency of computer processing of of statistical data controls of accuracy products of mechanical engineering. Presented by implementation at production of precision control of product quality with the use of of computer facilities for the processing of the measurement results..

Keywords: control, accuracy, product, static information, program.

Вступ. Успішний розвиток багатьох галузей машинобудівної промисловості пов'язаний з використанням великої кількості підшипників кочення, що виготовляються в умовах серійних виробництв. Часто якість функціонування цих машин залежить від експлуатаційних характеристик підшипникових вузлів, а показники якості прямо залежать від експлуатаційних характеристик підшипників. Через це вимоги до якості виготовлення підшипників і до їх експлуатаційних показників постійно зростають, а технологічне забезпечення цих вимог стає все проблематичнішим [1]. Саме якість продукції є одним з найважливіших засобів конкурентної боротьби, завоювання і утримання позицій на ринку. Тому підприємства приділяють особливу увагу забезпеченню високої якості продукції [2]. Одним з таких факторів підвищення якості вироблених деталей є своєчасне усунення похибок під час обробки [3]. У цих умовах актуальною є проблема виявлення резервів підвищення точності оброблюваних поверхонь. Вирішується ця проблема за допомогою впровадження аналітичної обробки за допомогою програмного забезпечення для управління якістю, виявлення та уникнення похибок під час виготовлення різноманітних виробів.

Мета роботи. Мета дослідження – аналіз точності обробки виготовлення виробів у вигляді тонкостінних кілець з використанням програмного продукту Minitab статистичної обробки даних.

Виклад основного матеріалу. Аналітична обробка інформації дуже трудомістка сама по собі і вимагає великого обсягу різноманітних обчислень. З переходом до ринкових відносин потреба в аналітичній інформації значно збільшується. У зв'язку з цим автоматизація аналітичних розрахунків стала об'єктивною необхідністю, обумовленою зростанням значення якісного інформаційного обслуговування процесу виготовлення виробів, стрімким розвитком технічних можливостей сучасних комп'ютерних технологій. Застосування комп'ютерних технологій підвищує ефективність аналітичної роботи, що досягається за рахунок скорочення термінів проведення аналізу, більш повного охоплення впливу факторів на результати обробки, заміни наближених чи спрощених розрахунків точними обчисленнями, що потребують значних затрат у часі в разі використання традиційних методів. Обробка статистичних та дослідних даних полягає у систематизації та встановленні якісних і кількісних залежностей між факторами, що досліджувались. Їх оцінка, зведення та інтерпретація проводиться на основі методів математичної статистики [4]. В сучасних умовах на шляху комп'ютеризації та інформатизації всіх процесів математична обробка результатів дослідження точності механічної обробки ведеться на основі комп'ютерних пакетів статистичного аналізу. Набір комп'ютерних систем та статистичних програмних продуктів на світовому ринку дуже великий і різноманітний. Добираючи програмне забезпечення, необхідно звернути увагу на деякі важливі характеристики, такі як придатність

звітних форм та графічна інтерпретація отриманих даних. У рамках застосування комп'ютерних технологій використовується інформаційний фонд підприємства, який представлений у формі баз даних і програмних засобів. Бази даних являють собою фактографічні дані певного процесу на виробництві. Програмні засоби утворюють інструмент автоматизованого виконання аналітичних задач для інформаційного обслуговування виробництва.

Для підвищення точності обробки різанням в технології машинобудування важливим елементом є дослідження технологічних процесів обробки поверхонь деталей. Найбільше застосування отримав статистичний метод, що має невисоку вартість і трудомісткість та дає можливість встановити умови оптимального функціонування досліджуваного процесу. Статистичний метод ґрунтується на отриманні та обробці великої кількості експериментальних даних, які забезпечують необхідний обсяг інформації. Його застосовують для дослідження точності технологічних процесів у серійному і масовому виробництвах [5]. Для реалізації статистичного методу на виробництві широко застосовують програми: Statistica; SPSS; SAS; Statgraphics+; Minitab; Systat; Stadia; Excel, тощо.

Найбільш кращим для вирішення задач точності обробки кільцевих заготовок є програмний продукт Minitab [6]. Його використовують п'ятсот найбільших світових корпорацій у 80 країнах світу. Ключові особливості програми: проста система аналізу даних, зокрема, Мастер Minitab Assistant допомагає аналізувати та інтерпретувати дані; наявність функції для описової статистики, перевірки гіпотез, довірчих інтервалів та критеріїв нормальності процесів; виявлення важливих факторів, що впливають на якість виробів; створення планів вибіркового контролю. Minitab має широкий спектр різних статистичних інструментів для проведення любых необхідних досліджень. Таких, як створення абсолютно любого графіка (рис.1), налаштування його кожного елементу під конкретні експериментальні дані, автоматичне оновлення графіків під час зміни даних.

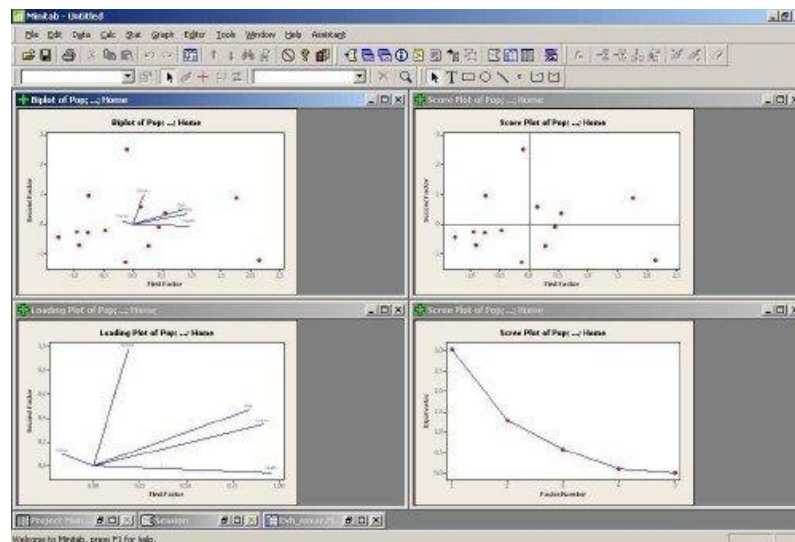


Рис.1.

Для роботи з використанням комп'ютерної програми MINITAB необхідним є заповнення:

- для карти медіан $\bar{X}$ – карти розмахів R (рис. 2);
- для карти індивідуальних значень $\bar{X}$ – карти змішуваних розмахів mR (рис. 3).



Рис. 2. Послідовність заповнення: карта медіан $\tilde{X}$ – карта розмахів R

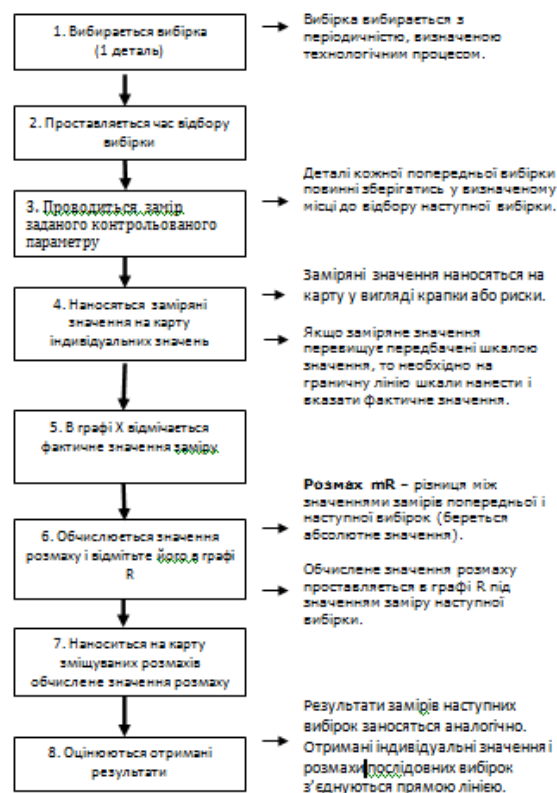


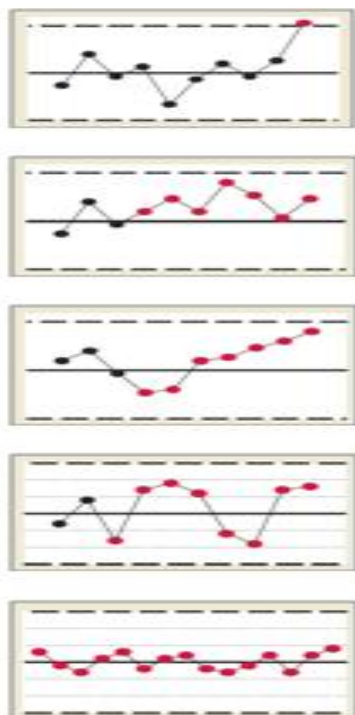
Рис. 3. Послідовність заповнення: карта індивідуальних значень X – карта зміщуваних розмахів mR

Дослідження проводилося у виробничих умовах згідно з інструкцією підприємства АТ «СКФ Україна» (публічне) по оцінюванню можливостей обладнання та процесів. Відповідно до цієї інструкції, проведення експериментальних досліджень організовує технолог відповідного виробничого підрозділу. Перед проведенням дослідження технолог виробничого підрозділу перевіряє: відповідність технічного стану обладнання паспортним даним; відповідність технічного оснащення конструкторській документації; відповідність встановлених режимів обробки технологічному процесу; відповідність засобів вимірювальної техніки: вимогам технологічного процесу (модель, вимірювальна головка), терміну придатності, відхиленню контрольованих характеристик. Збір даних проводиться технологом виробничого підрозділу, який відбирає деталі в порядку обробки без втручання в процес. Технолог виробничого підрозділу відбирає вибірки: послідовно, через визначені інтервали часу; з однаковою кількістю деталей; з охопленням якомога більшої кількості особливих причин мінливості. Вимірювальна система, яка використовується для проведення замірів, повинна бути прийнятною. Технолог виробничого підрозділу проводить заміри характеристик деталей вибірок, дотримуючись послідовності їх відбору і заносить всі значення в контрольний лист.

Для подальшого аналізу даних додатково фіксується:

- час початку та закінчення збору даних;
- кожне втручання в процес обробки деталей;
- всі зупинки в роботі обладнання;
- зміна (зміни) /оператора;
- непередбачувані перерви в процесі обробки;
- зміна партії заготовок.

Всі деталі, які використовуються для проведення вивчення можливостей зберігаються до завершення проведення статистичної оцінки. Величина вибірки (кількість відібраних для замірів деталей) повинна становити не менше 125 штук. За даними досліджень будуються графіки, один з яких будується за даними різниці між значеннями попередньої та наступної вибірок (партія з 5 послідовно заміряних деталей), береться абсолютне значення. У іншому графіку, загальна кількість замірів ділиться теж на вибірки по 5 деталей, але з кожної вибірки береться різниця між максимальним та мінімальним значенням замірів (рис. 4).



- крапка за межами контрольних меж

- сім послідовних крапок вище або нижче середньої лінії

- тенденція до зростання або спадання семи послідовних крапок

- вісім послідовних крапок наближені до контрольних меж

- п'ятнадцять послідовних крапок наближені до середньої лінії

Рис. 4.

Для визначення величини деформації деталі в процесі обробки було прийнято дослідити точність форми кільця однорядного роликового конічного підшипника 32017X по зовнішньому

діаметру (рис. 5, а). Досліджувальний параметр відхилення форми не круглість, а саме її частковий випадок огранка. Огранка виникає після проведення обточування кільця під час його затиснення у токарному патроні. Для контролю застосовували найбільш простий метод, що гарантує достовірність вимірювання, за допомогою індикатора годинникового типу і установкою деталі в призмі (рис. 5,б).

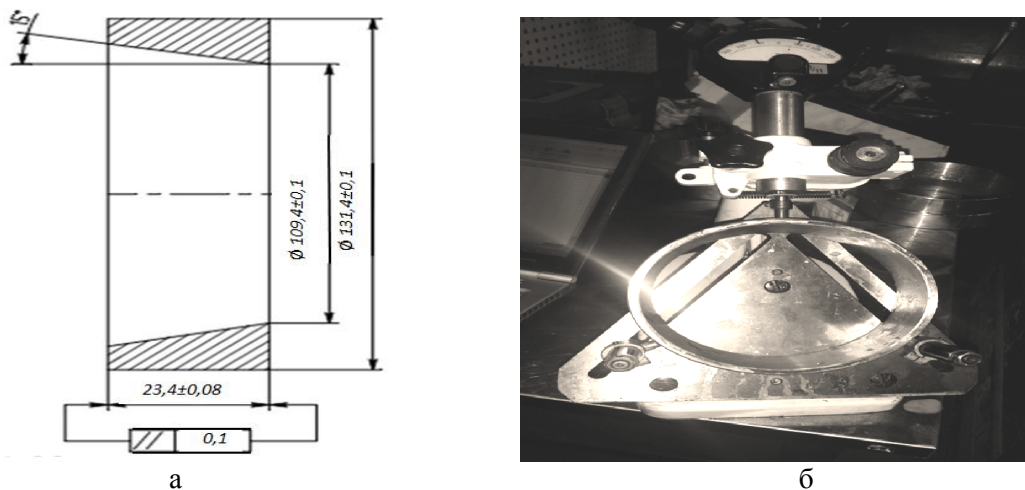
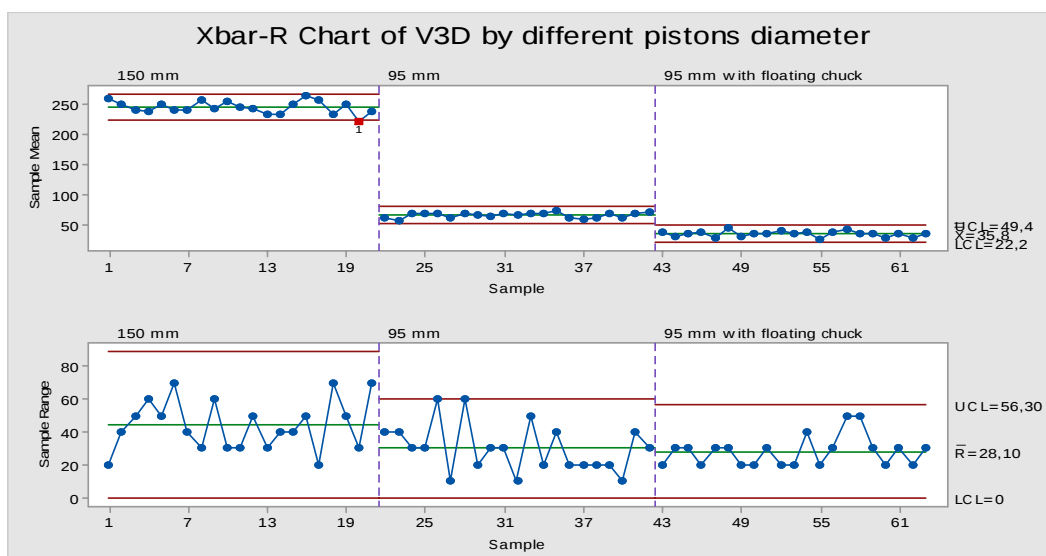


Рис.5. Кільце підшипника 32017Х: а – макет заготовки кільця; б – контроль кільця

Деталі для дослідження вибиралися безпосередньо у виробничому підрозділі – дільниці токарної обробки АТ «СКФ Україна» (публічне) в порядку обробки без втручання в процес. Всі деталі оброблялися за однакових режимів обробки (подача – 0,3мм/об, частота обертання шпинделя верстата – 376 об/хв,) , без охолодження і з однаковим зусиллям затиску в патроні (тиск в гідросистемі верстата становить 10...15 кг/см²).

Досліджувалися вибірки з однаковою кількістю деталей – 130 штук в кожній. Деталі відбиралися після токарної обробки із затисканням у 3-х кулачковому патроні з пневмоциліндром діаметрами поршнів Ø 150 мм і Ø95 мм та плаваючому 6-ти кулачковому патроні – пневмоциліндром діаметром поршня Ø95мм. Вимірювальна система, яка використовувалася для проведення замірів, була прийнятною. Заміри характеристик деталей вибірок проводилися у послідовності їх відбору. Всі деталі, які досліджувалися зберігалися до завершення проведення статистичної оцінки. Після замірів, проводили розрахунки за допомогою комп'ютерної програми Minitab. Результати статистичної оцінки представлені у вигляді графіків ймовірності (рис. 6).



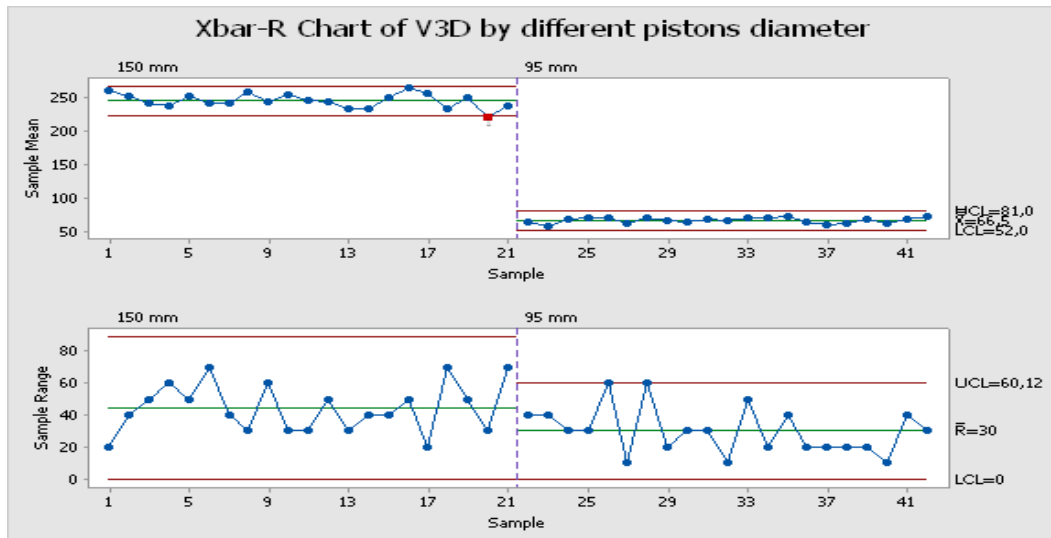


Рис. 6. Графіки ймовірності для дослідження огранки

Фактичні відхилення контрольованого параметру максимально наближені до середини поля допуску. Однак є випадки виявлення незвичайної поведінки кривої (як на карті розмахів R/ зміщуваних розмахів mR, так і на карті медіан X/ індивідуальних значень X), що характеризується одним або декількома критеріями статистичної некерованості процесу. Так наприклад, на рис. 7 є випадки коли заміряне значення перевищує передбачені шкалою значення тому необхідне негайне втручання в технологічний процес.

Висновки. Якість продукції формується під впливом використання сучасних комп'ютерних технологій, як на усіх етапах виготовлення так і контролю деталей особливо для умов гнучких автоматизованих виробництв машинобудування, зокрема виробництва підшипників. Перевірка оператором чи налагоджувальником параметрів точності деталей, що послідовно обробляються необхідна для підтвердження правильності налагодження верстата. Фактичні відхилення контрольованих параметрів повинні бути максимально наближені до середини поля допуску. Якщо результати перевірок не відповідають встановленим вимогам, то необхідно визначити причини невідповідностей та розробити необхідні коригувальні дії. На виробництві для цього розробляється та реалізується план реагування за результатами статистичної оцінки, в якому ідентифікуються і вказуються засоби усунення особливих причин: технологічні можливості обладнання, засоби вимірювальної техніки, технологічний процес, а також джерело мінливості, що впливає лише на деякі виходи процесу, найчастіше діє на процес стрибкоподібно та непередбачуваною.

1. Марчук В.І., Михалеви́ч В.Т. Відхилення форми поверхонь та їх вплив на розмірну точність тіл обертання// Наукові нотатки: міжвузівський збірник (за напрямом "Інженерна механіка") – Луцьк: Луцький державний технічний університет, - 2008. Випуск 23. – с.197-202.
2. Шевчук Д.А. Контроль якості / Д.А. Шевчук – М:Гросс-Медіа, 2009. – 286с.
3. Петріков А.В., Іванов О.О. Види і причини похибок при механічній обробці деталей, а також методи їх усунення [Електронний ресурс] - А.В. Петріков, О.О. Іванов // Scientific researches and their practical application. Modern state and ways of development '2012. – 2-12 October 2012. – Режим доступу: <http://www.sworld.com.ua/index.php/ru/conference/the-content-of-conferences/archives-of-individual-conferences/oct-2012>. – Назва екрану.
4. Айвазян С.А., Степанов В.С. Інструменти статистичного аналізу даних / С.А. Айвазян, В.С. Степанов - М.: Наука, 2000. – 302с.
5. Клячкин В.Н. Технология многомерного статистического контроля процесса / В.Н. Клячкин // Информационные технологии в проектировании и производстве. – 2002. – № 1. – С. 49-53.
6. Statgraphics Centurion XVI.I + Minitab 16 [Електронний ресурс] – Режим доступу: <http://mistergid.ru/pc/soft/36306-statgraphics-centurion-xvii-minitab-16.html>. – Назва екрану.

УДК 378:004.9

Горбатюк Р. М., д. пед. н., професор

Тернопільський національний педагогічний університет ім. В. Гнатюка

ПЕДАГОГІЧНІ АСПЕКТИ ПІДГОТОВКИ ІНЖЕНЕРІВ-ПЕДАГОГІВ ЗАСОБАМИ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Горбатюк Р. М. Педагогічні аспекти підготовки інженерів-педагогів засобами інформаційно-комунікаційних технологій. Розкриваються особливості застосування інформаційно-комунікаційних технологій у підготовці майбутніх інженерів-педагогів. Установлено, що для ефективного формування їх професійних компетентностей необхідно використати комп'ютерне моделювання.

Ключові слова: технології, інженер-педагог, навчання, компетентності, інтеграція, моделювання.

Горбатюк Р. М. Педагогические аспекты подготовки инженеров-педагогов средствами информационно-коммуникационных технологий. Раскрываются особенности применения информационно-коммуникационных технологий в подготовке будущих инженеров-педагогов. Определено, что для эффективного формирования их профессиональных компетентностей необходимо использовать компьютерное моделирование.

Ключевые слова: технологии, инженер-педагог, обучение, компетентности, интеграция, моделирование.

Horbatyuk R. M. Pedagogical aspects of engineers-teachers preparation by means of information and communication technologies. The peculiarities of computer-cognitive technologies' application due to preparation of future engineer-teachers are established. Besides, it is necessary to use computer design in order to effective forming their professional abilities.

Keywords: technologies, engineer-teacher, training, competence, integration, modeling.

Постановка наукової проблеми. Розвиток комп'ютерної техніки загострив перед освітньою галуззю багато проблем, однією з яких є використання у навчальному процесі сучасних інноваційних технологій. В умовах становлення інформаційного суспільства головним завданням вищої школи є створення стійкої мотивації до навчання, залучення студентів до самоосвіти, а освітній процес розглядається як засіб формування майбутніх фахівців, що спричинений розвитком їх творчого мислення.

Спрямування освітньої системи на особистість, як головний соціальний орієнтир, проявляється в різних напрямках, і провідним серед них є створення для кожного члена суспільства можливості отримати освіту будь-якого профілю та рівня впродовж усього життя. Становлення особистісно-орієнтованої системи освіти неможливе без підготовки для неї фахівців нового покоління, здатних у своїй професійній діяльності реалізовувати нову освітню парадигму. Тому на сучасному етапі інформатизації вищої освіти пріоритет належить загальнотеоретичним, фундаментальним та інтегративним знанням інформаційно-комунікаційних технологій [1; 5].

Поступове економічне зростання, що відбувається останніми роками в Україні, вимагає від освітньої системи підготовки кваліфікованих фахівців, здатних конкурувати на ринку праці. У період економічного становлення держави значної підтримки потребує підготовка майбутніх інженерів-педагогів в галузі комп'ютерних технологій, тому що у системі професійної освіти виникають питання щодо її забезпеченості викладацькими кадрами. Часто інженерні дисципліни викладають фахівці, які мають недостатню професійну кваліфікацію (викладачами комп'ютерних дисциплін працюють інженери-програмісти, математики, фізики, інженери інших спеціальностей). Така тенденція має місце і серед технічного персоналу (майстрів виробничого навчання, лаборантів). Важливістю цієї проблеми є те, що професійна діяльність майбутніх інженерів-педагогів в галузі комп'ютерних технологій буде здійснюватися за допомогою інформаційно-комунікаційних технологій (ІКТ).

Аналіз досліджень. Використання у навчальному процесі інформаційно-комунікаційних технологій розглядається у працях В. Бикова, Т. Волкова, М. Жалдака, Н. Морзе, Н. Ничкало, А. Нісімчука, Ю. Триуса, О. Щербак і ін.

Проблему професійної підготовки фахівців в галузі комп'ютерних технологій на основі інформаційно-комунікаційних технологій у педагогіці розглянуто в докторських і кандидатських дисертаціях К. Акуленко, Р. Гуревича, Є. Громова, В. Кабака, О. Потапчук та ін. Проте ці питання досліджуються науковцями в загальнотеоретичному плані, без урахування вимог конкретної професійної діяльності майбутніх інженерів-педагогів.

Формулювання цілей статті. Метою статті є визначення концептуальних аспектів реалізації сучасних інформаційно-комунікаційних технологій у підготовці майбутніх інженерів-педагогів в галузі комп'ютерних технологій в умовах педагогічного університету.

Виклад основного матеріалу й обґрунтування отриманих результатів. Підготовка майбутніх інженерів-педагогів в галузі комп'ютерних технологій у вищих навчальних закладах інженерно-педагогічного, педагогічного і технічного спрямування, як відносно новий вид підготовки інженерно-педагогічних фахівців для системи професійної освіти у ВНЗ I-II рівнів акредитації перебуває на етапі активного становлення. Це передусім стосується змісту професійної і практичної їх підготовки у ВНЗ. Окрім цього, в інженерно-педагогічних і технічних ВНЗ інженерна підготовка за обсягом здебільшого є ґрунтовнішою порівняно з педагогічною, що призводить до зміни спрямування підготовки таких фахівців [3].

Специфіка кваліфікації «інженер-педагог» є досить чітко окресленою в освітньо-кваліфікаційній характеристиці (ОКХ), проте продовжує викликати дискусію її раціональність: за п'ять років підготувати інженера і педагога складно, проте можливо за умов використання нових форм, методів, засобів навчання і, відповідно, сучасних інформаційно-комунікаційних технологій.

Враховуючи інтегральний характер професійної діяльності майбутніх інженерів-педагогів, спираючись на досвід роботи в Тернопільському національному педагогічному університеті імені Володимира Гнатюка (ТНПУ), можна передбачити необхідні методи навчання засобами інформаційно-комунікаційних технологій. На нашу думку, у процесі підготовки таких фахівців, доцільно використати комп'ютерне моделювання.

Для підготовки здобувачів вищої освіти першого (бакалаврського) рівня спеціальності 015 Професійна освіта (Комп'ютерні технології) вагоме значення має графічний спосіб подання матеріалу за допомогою методу моделювання [10, с. 214]. З його використанням дослідження в предметі певних ознак відбувається ефективніше.

Суть комп'ютерного моделювання полягає в отриманні якісних і кількісних результатів за допомогою фізичної або комп'ютерної моделі. Якісні показники дозволяють виявити властивості системи професійної підготовки майбутніх інженерів-педагогів, зокрема: структуру, динаміку розвитку, стійкість, цілісність тощо. Кількісні показники мають прогностичний характер щодо її (системи) ефективності. Цілі комп'ютерного моделювання можуть бути різними, проте найчастіше моделювання є основною процедурою системного аналізу, під яким розуміємо сукупність методологічних засобів, які використовуються для підготовки та прийняття рішень різного характеру. На нашу думку, комп'ютерна модель складної системи повинна відображати основні чинники та взаємозв'язки, які характеризують реальні об'єкти, ситуації, критерії. Вона (комп'ютерна модель) повинна бути універсальною, щоб забезпечити опис наближених за значенням об'єктів-оригіналів і виконати необхідні дослідження з мінімальними затратами [3, с. 45].

Комп'ютерні моделі легко вписуються в заняття: дозволяють педагогу продемонструвати більшість властивостей об'єкта, організувати нові, нетрадиційні види навчальної діяльності тощо. У процесі підготовки майбутніх інженерів-педагогів метод моделювання є основним інструментом пізнання педагогічних явищ, технологічних і виробничих процесів у тих випадках, коли провести педагогічний експеримент в реальних умовах неможливо. Евристична сила методу моделювання визначається тим, що з його допомогою вдається звести складне до простого, тобто зробити предмет доступним для ретельного та всебічного вивчення [2, с. 114–115].

Враховуючи сказане вище, одним із засобів активізації пізнавальної діяльності майбутніх інженерів-педагогів під час вивчення дисциплін професійної та практичної підготовки є їх комп'ютерна підтримка, яка з використанням педагогічно виважених і доцільно дібраних програмних засобів дає значний педагогічний ефект.

Вважаємо, що формування інформаційно-технічної підготовки майбутніх інженерів-педагогів у процесі вивчення дисциплін професійної та практичної підготовки закладає підґрунтя для сприйняття нового навчального матеріалу на основі інтегративного підходу, дає орієнтири студентам у значній кількості педагогічних, технічних і виробничих знань, котрі швидко оновлюються на основі фундаментальних знань в галузі інформаційно-комунікаційних технологій. Уточнення змістової частини, відбір додаткових фактів, важливо для освоєння основ вибраної професії.

Враховуючи інтегральний характер професійної діяльності майбутніх інженерів-педагогів, спираючись на власний досвід роботи в педагогічному університеті, передбачили необхідні методи навчання, що забезпечувалися відповідними засобами інформаційно-комунікаційних технологій. Оскільки покращення якості професійної підготовки інженерів-педагогів тісно пов'язане з підвищенням ефективності педагогічної системи ВНЗ, то в процесі підготовки

майбутніх інженерів-педагогів до професійної діяльності використали, крім традиційних методів навчання (вербальні, практичні), комп'ютерне моделювання.

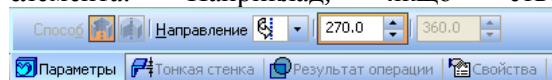
Розглянемо приклад вивчення теми «Основи 3D-моделювання» (предмет «Комп'ютерна графіка») на основі інтегративного підходу і методу комп'ютерного моделювання.

Аналіз алгоритмів створення графічних зображень засобами комп'ютерної техніки є підготовчим етапом засвоєння знань з курсу «Інженерна комп'ютерна графіка». Вибір інструментарію для побудови зображень у двовимірному просторі, оптимальна послідовність дій, перетворення створених зображень в об'ємні твердотільні моделі, формує в майбутніх фахівців в галузі комп'ютерних технологій розуміння принципів 3D-моделювання. Надзвичайно важливим у цьому випадку є процес запам'ятовування, що реалізується під час відтворення навчального матеріалу і спричинює мнемічні дії, стаючи наскрізною лінією структури досвіду майбутнього інженера-педагога.

У всіх сучасних системах тривимірного моделювання побудова твердотільної моделі відбувається за загальним принципом, що передбачає послідовне виконання булевих операцій (поєднання, віднімання, перетину) над об'ємними елементами (призмами, циліндрами, пірамідами, конусами тощо).

Виходили з того, що сучасні 3D-системи, зокрема графічний редактор КОМПАС 3D, володіють ефективними засобами моделювання, які дозволяють створювати тривимірні моделі складних моделей і зборок [6, с. 192–193]. Використовуючи наочні методи створення об'ємних елементів, оперували простими та чіткими поняттями: основа, бобишка, фаска, заокруглення, отвір, ребро, оболонка тощо. Крім цього, процес проектування часто відтворює технологічний процес виготовлення деталі.

На розвиток мислення були спрямовані питання проектування технологічних процесів виготовлення деталей. Створення об'ємних елементів передбачає переміщення плоских фігур у просторі. У процесі переміщення ці фігури обмежували частину простору, визначаючи форму елемента. Наприклад, якщо створений ескіз перемістити на кут 270°



навколо осі, що лежить у його площині, то одержимо тривимірний твердотільний об'єкт із відповідним кутом (рис. 1).

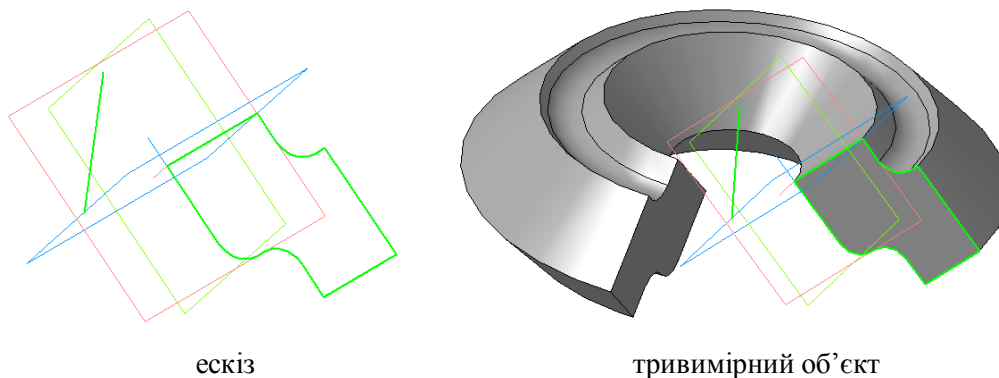


Рис. 1. 3D-об'єкт, створений методом Обертання

Розуміння наочних методів створення об'ємних тіл формує в майбутніх інженерів-педагогів базові знання про ефективні способи моделювання, що дозволило створювати 3D-моделі складних за формою деталей і зборок.

Залежно від контингенту студентів викладач може вибрати у міжпредметному модулі ті питання та відомості, що є вагомими. Для вибору пропонували в кожному модулі від трьох до п'ятдесяти питань. Наприклад: загальні принципи моделювання деталей; основні терміни тривимірної моделі (грань, вершина, тіло деталі); формоутворювальні операції; послідовність створення ескізу основи; вимоги до ескізів: основні підходи до формування тривимірних формоутворювальних елементів у твердотільному моделюванні (видавлювання, обертання, кінематична операція, операція за січеннями). Під час розв'язання зазначених завдань у студентів формуються елементи творчого мислення шляхом аналізу-синтезу, порівняння, абстрагування,

конкретизації. Їх сукупність зумовлює формування творчого мислення, образних і наочно-практичних компонентів, що вкрай необхідні майбутнім інженерам-педагогам для вирішення комплексу інженерних завдань засобами інформаційно-комунікаційних технологій [4; 8].

Використання у навчальному процесі засобів ІКТ забезпечило можливість систематично розглядати різні способи розв'язування завдань, збільшити їх кількість, урізноманітнити зміст, розширити можливості узагальнень інженерно-педагогічних понять. У процесі розв'язування завдань професійного спрямування засобами комп'ютерної техніки у студентів на якісно новому рівні формується культура розумової праці, а також такі важливі вміння як: планувати свою роботу, раціонально її виконувати, критично зіставляти початковий стан роботи з реальним процесом її виконання. Прикладом цього є поетапний процес створення тривимірних моделей, який передбачає:

- вибір проекційної площини – забезпечується виключним розумінням конструкції майбутньої деталі та особливостями розташування у складальному вузлі;

- побудову ескізу, який повинен бути інформативним (максимально відображати форму предмета) і раціональним з точки зору конструктивних особливостей. Досягнути цього можна шляхом застосування визначених алгоритмів нанесення необхідних розмірів і геометричних залежностей, на основі сформованих графічних компетентностей (курс «Інженерна комп'ютерна графіка»);

- формування основи просторової моделі – забезпечується розумінням і вмінням застосувати елементний інструментарій для побудови базової геометричної форми виробу;

- створення конструктивних елементів (формування на базовій основі основних категорій модифікацій). Студенти заздалегідь повинні проаналізувати створювану віртуальну модель, спрогнозувати доцільну кількість операцій і розуміти технологію виготовлення виробу в реальному вимірі.

Розроблення тривимірної моделі – складний творчий процес, що вимагає від майбутніх інженерів-педагогів не тільки знань основ проектування та програмних засобів, а також неординарного і гнучкого мислення. Зокрема, надзвичайно важливе значення має вибір раціонального способу одержання деталі у виробничих процесах. З такими явищами майбутні фахівці в галузі комп'ютерних технологій зустрічаються, коли ознайомлюються з різними технологіями, вивчають дисципліни професійної та практичної підготовки («Комп'ютерні технології в навчальному процесі», «Імітаційне моделювання технічних систем», «Комп'ютерний дизайн та мультимедіа» тощо).

Зазначене вище переконливо доводить доцільність застосування методу комп'ютерного моделювання у навчальному процесі підготовки майбутніх інженерів-педагогів. Цей метод дозволив цілісно вивчати досліджуваний об'єкт, оскільки з'явилася можливість відтворити аналог реально існуючих або спроектованих дослідником систем і процесів, виявити не тільки структуру, властивості, елементи, а також взаємозв'язки між ними. Використання моделей з різних предметних галузей дозволило реалізувати інтегративний підхід до вивчення дисциплін професійної та практичної підготовки, що важливо для майбутньої професійної діяльності інженерів-педагогів в галузі комп'ютерних технологій. Працюючи за комп'ютером, вони виконують різні вправи (на основі індивідуального підходу) щодо побудови моделей, вузлів, механізмів, удосконалюючи свої навички, доводячи їх до автоматизму. Робота за комп'ютером формує мотивацію студентів до навчання, сприяє активному включенню в процес пізнання. Використання комп'ютерних засобів дає можливість викладачу під час занять акцентувати увагу на стимулюванні процесів саморозвитку студентів. Водночас у педагога з'являється можливість поширити контроль за засвоєнням знань й управляти цим процесом [7; 9].

Одним із напрямів підвищення рівня ефективності вивчення дисциплін професійної та практичної підготовки є педагогічно вивірене використання інформаційно-комунікаційних технологій у поєднанні з системою психолого-педагогічних засобів активізації навчальної діяльності. Застосування комп'ютерних засобів передбачає опанування студентами психолого-педагогічних знань і навичок використання методів навчання, прикладного програмного забезпечення, тобто використання сучасних інформаційно-комунікаційних технологій, доступ до яких забезпечує навчальний заклад. Використання комп'ютерних засобів дає можливість під час вивчення теоретичного матеріалу звернути основну увагу студентів на з'ясування суті досліджуваних явищ, побудову імітаційних моделей, інтерпретацію результатів, отриманих за допомогою комп'ютера, зекономити час на побудову графічних зображень тощо.

Сучасні тенденції застосування інформаційно-комунікаційних технологій у навчальному процесі свідчать про те, що для досягнення успіху майбутній інженер-педагог в галузі комп'ютерних технологій повинен однаково добре орієнтуватися: у самому об'єкті, процесі, системі проектування; в апараті обробки та аналізу вхідної і вихідної інформації про об'єкт, процес, систему, зовнішнє середовище; у математичному моделюванні, тобто в постановці та формалізації завдання, яке полягає в умінні перевести технічне завдання з проблемно-змістового на мову математичних схем і моделей і далі в спеціальне програмне забезпечення; у методах пошуку оптимального рішення; у відповідному програмному забезпеченні систем автоматизованого проектування (діалогових системах, банках даних, базах знань та ін.); у вільному володінні засобами обчислювальної техніки.

Висновки та перспективи подальших досліджень. Таким чином, система професійної освіти забезпечує вирішення множинних проблем, головними серед яких є питання інтеграції психолого-педагогічних і комп'ютерних знань у змісті підготовки майбутніх інженерів-педагогів в галузі комп'ютерних технологій. Поєднання такої підготовки має взаємообумовлений характер: з одного боку – у вивченні циклу психолого-педагогічних дисциплін та їх забезпечення змістовною комп'ютерною підготовкою; з іншого – психолого-педагогічна підготовка є одним із основних детермінантних чинників відбору змісту циклу комп'ютерних дисциплін і практик.

Удосконалення змісту підготовки майбутніх інженерів-педагогів в галузі комп'ютерних технологій є подальшим кроком досліджень.

1. Биков В. Ю. Моделі організаційних систем відкритої освіти : монографія / Валерій Юхимович Биков. – Київ : Атіка, 2009. – 684 с.
2. Волкова Т. В. Інтеграція педагогічної та комп'ютерно-інформаційної підготовки майбутнього викладача спеціальних дисциплін професійно-технічного навчального закладу : дис. ... канд. пед. наук : 13.00.04 / Волкова Тетяна Василівна. – Київ, 2007. – 304 с.
3. Горбатюк Р. М. Підготовка майбутніх інженерів-педагогів до професійної діяльності засобами комп'ютерних технологій : монографія / Р. М. Горбатюк, В. В. Кабак. – Луцьк : ВМА «Терен», 2015. – 264 с.
4. Громов Є. В. Формування педагогічних знань і вмінь майбутніх інженерів-педагогів у процесі навчання комп'ютерних дисциплін : дис. ... канд. пед. наук : 13.01.02 / Громов Євген Володимирович. – Харків, 2006. – 248 с.
5. Гуревич Р. С. Інформаційно-комунікаційні технології в навчальному процесі і наукових дослідженнях / Р. С. Гуревич, М. Ю. Кадемія. – К. : Освіта України, 2007. – 396 с.
6. Ефремов Г. В. Инженерная и компьютерная графика на базе графических систем: учебное пособие / Г. В. Ефремов, С. И. Ньюкалова. – Издательство: Тонкие наукоемкие технологии, 2014. – 256 с.
7. Жалдак М. І. Інформаційні технології : навч.-метод. посіб. / [М. І. Жалдак, О. А. Хомік, І. В. Володько, О. М. Снігур] ; за ред. М. І. Жалдака. – К. : РННЦ «ДІНІТ», 2003. – 194 с.
8. Кабак В. В. Дослідження ефективності методики формування готовності майбутніх інженерів-педагогів до використання комп'ютерних технологій / Л. М. Романишина, В. В. Кабак // Оновлення змісту, форм та методів навчання і виховання в закладах освіти : збірник наукових праць / редкол. : Бех І. Д. [та ін.]. – Рівне : РДГУ, 2013. – Вип. 7 (50). – С. 146–150.
9. Потапчук О. І. Застосування інформаційно-комунікаційних технологій в професійній підготовці інженерів-педагогів / О. І. Потапчук // Сучасні інформаційні технології та інноваційні методики навчання в підготовці фахівців: методологія, теорія, досвід, проблеми : збірник наукових праць Вінницького держ. пед. ун-ту ім. М. Коцюбинського ; за ред. І. А. Зязюна. – Київ ; Вінниця, 2015. – Вип. 42 – С. 331–335.
10. Шеннон Г. Р. Имитационное моделирование систем – искусство и наука : [пер. с англ.] / Г. Р. Шеннон ; под ред. Е. К. Масловского. – М. : Мир, 1978. – 418 с.

УДК 519.683.8

Горшунов А.І.

Білоруський державний університет інформатики і радіоелектроніки

АРХИТЕКТУРНАЯ МОДЕЛЬ ПОСТРОЕНИЯ ВЕБ-ПРИЛОЖЕНИЙ CQRS

Горшунов А.І. Архітектурна модель побудови веб-застосунків CQRS. У статті пропонується базова імплементація архітектурної моделі побудови застосунків за методологією CQRS (Command and Query Responsibility Segregation, поділ відповідальності на команди і запити), аналізуються особливості проектування, процесу розробки бізнес-додатків; переваги і недоліки даної методології.

Ключові слова: архітектура, процес розробки, CQRS, C#, програмне забезпечення.

Harshunou A.I. CQRS as architectural model web applications development. The article proposes basic implementation of architectural model building applications using CQRS methodology (Command and Query Responsibility Segregation), analyzes the characteristics of design, business application development process; the advantages and disadvantages of this methodology.

Keywords: architecture, development process, CQRS, C#, software.

Горшунов А.И. Архитектурная модель построения веб-приложений CQRS. В статье предлагается базовая реализация архитектурной модели построения приложений по методологии CQRS (Command and Query Responsibility Segregation, разделение ответственности на команды и запросы), анализируются особенности проектирования, процесса разработки бизнес-приложений; преимущества и недостатки данной методологии.

Ключевые слова: архитектура, процесс разработки, CQRS, C#, программное обеспечение.

Постановка проблемы. Процесс разработки программного обеспечения изменяется с развитием технологий программирования: системы усложняются, количество данных и пользователей растёт, изменяется и процесс разработки непосредственно программистами.

Заказчики приложений редко могут дать подробное описание всех бизнес-правил и функционала приложений, т.к. сами зачастую не знают всех требований и ожиданий клиента, формируя «сырое» техническое задание и перекладывая ответственность и выяснение требований непосредственно на команду разработчиков во время этапа разработки.

Основной проблемой разработчиков становится поддержка постоянно меняющихся бизнес-правил, интеграций, оптимизация производительности и расширение функциональности приложений, и, как следствие, проблема качества кода.

В крупных бизнес-приложениях зачастую существует множество предметных областей и сложных правил, которые сложно спроектировать и описать единой моделью. Зачастую бизнес-аналитики сами не в состоянии описать модель и выявить сущности приложения либо провести декомпозицию. В итоге аналитика и моделирование предметной области частично (а иногда и полностью) перекладывается на плечи разработчиков.

Также сами разработчики могут понимать и разделять приложение на компоненты и сущности по-разному, создавая тем самым проблемы дублирования кода, запросов, интерфейсов и даже перемешивание слоёв приложения, инфраструктурного кода и кода доменной модели.

Таким образом, учитывая особенности процесса разработки бизнес-приложений, разработчикам необходимо спрогнозировать возможность изменений, унифицировать терминологию и построить архитектуру приложения так, чтобы минимизировать «архитектурные» риски (вероятность полного переписывания приложения) в связи с изменившимися бизнес-правилами.

Целью работы является изучение и демонстрация возможностей применения архитектурной модели построения бизнес-приложений CQRS.

Основные результаты исследования. Модель Command-Query Separation (CQS) предлагает разделение работы с объектом на Запросы (Query) и Команды (Commands). При этом необходимо соблюдать следующие правила:

- запросы возвращает данные и никогда не меняют состояние объекта;

- команды изменяют состояние объекта, в идеальном случае, не должны ничего возвращать.

Из этого следует, что во время отсутствия Команд:

- одинаковые Запросы гарантировано вернут одинаковый результат;
- любое количество любых Запросов не изменят состояние объекта;
- удаление Запроса из кода абсолютно прозрачно для объекта и не может дать побочных действий.

Эти свойства CQS используются в «программировании по контракту». В нем методы, осуществляющие проверку состояния объекта, являются Запросами. Это гарантирует их прозрачность для остального кода и возможность безопасного удаления части из них в Release версии кода (например, для оптимизации). Кроме того, подход, предлагаемый CQS, позволяет сделать код приложения более понятным именно благодаря разделению Команд и Запросов. Соответственно, в дальнейшем такое приложение легче поддерживать и модифицировать.

Необходимо отметить, что в некоторых случаях CQS не может быть использован в принципе. В качестве примеров можно рассмотреть реализации алгоритмов стек и очередь. Их методы Pop() и Dequeue() по определению должны и возвращать данные и изменять состояние самого объекта. Кроме того, в случае многопоточного программирования иногда удобнее иметь методы, противоречащие принципам CQS.

Область использования CQS достаточно широка. Например, он подходит для взаимодействия с элементами управления пользовательского интерфейса. Однако, в случае применения данного принципа для работы с источниками данных используется термин Command-Query Responsibility Segregation (CQRS). По сути это синоним, уточняющий область применения. CQRS несет в себе все принципы CQS и не ограничивает способ хранения данных. Это может быть СУБД, массив в памяти и т. д.

Таким образом, мы получаем набор объектов, которые меняют состояние системы, и набор объектов, которые возвращают данные. Типовой дизайн системы, где есть пользовательский интерфейс, бизнес-логика и база данных, показан на рисунке 1:

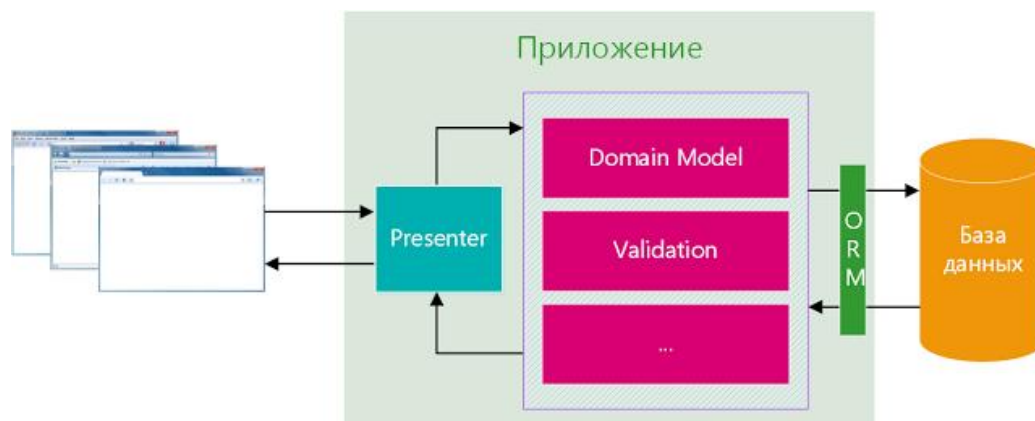


Рис.1. Классический пример архитектуры бизнес-приложения

CQRS предполагает, что не нужно смешивать объекты Команда и Запрос, а нужно их явно выделить (см. рисунок 2):

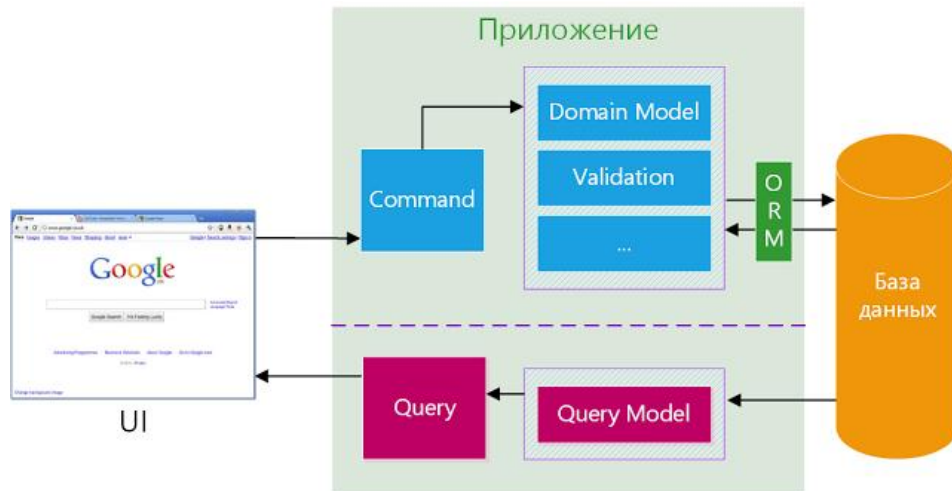


Рис. 2. Архитектурная модель CQRS

Дальше мы будем использовать эти базовые понятия и структуры, чтобы придать системе возможность горизонтального масштабирования. Но уже сейчас, явно выделив Команду и Запрос в вашем коде, вы получите более стройный дизайн системы. Если класс является Командой, то:

- изменяет состояние системы;
- ничего не возвращает;
- хорошо описывает предметную область, как действия пользователей над системой;
- контекст команды хранит нужные для её выполнения данные.

Если класс является Запросом, то:

- не изменяет состояние системы;
- контекст запроса хранит нужные для её выполнения данные;
- возвращает результат

В качестве демонстрации был выбран язык C#.

Основные интерфейсы в CQRS могут выглядеть так:

```
public interface IQuery<out TOutput>
{
    TOutput Ask();
}
public interface IQuery<in TSpecification, out TOutput>
{
    TOutput Ask(TSpecification spec);
}
public interface IAsyncQuery<TOutput> : IQuery<Task<TOutput>> { }
public interface IAsyncQuery<in TSpecification, TOutput> : IQuery<TSpecification, Task<TOutput>> { }
public interface ICommandHandler<in TInput>
{
    void Handle(TInput input);
}
public interface ICommandHandler<in TInput, out TOutput>
{
    TOutput Handle(TInput input);
}
public interface IAsyncCommandHandler<in TInput> : ICommandHandler<TInput, Task> { }
public interface IAsyncCommandHandler<in TInput, TOutput>: ICommandHandler<TInput, Task<TOutput>> { }
```

В этом случае в отсутствии Команд все Запросы всегда возвращают одинаковые результаты на одинаковых входных данных. Такая организация сильно упрощает отладку, потому что в Запросе нет состояния, которое могло бы изменить возвращаемый результат.

Приведем пример реального использования. Отчеты – не единственная частая задача чтения данных. Более общее определение типовых операций чтения это – фильтрация, постраничный вывод, создание проекций (представление агрегатов в необходимом на клиентской стороне виде). Для постраничного вывода из любой таблицы в базе данных и поддержкой фильтрации можно использовать всего одну реализацию IQuery:

```
public class ProjectionQuery<TSpecification, TSource, TDest>: IQuery<TSpecification,
IEnumerable<TDest>>
    , IQuery<TSpecification, int>
    where TSource : class, IHasId
    where TDest : class
{
    protected readonly ILinqProvider LinqProvider;
    protected readonly IProjector Projector;

    public ProjectionQuery(ILinqProvider linqProvier, IProjector projector)
    {
        if (linqProvier == null) throw new ArgumentNullException(nameof(linqProvier));
        if (projector == null) throw new ArgumentNullException(nameof(projector));
        LinqProvider = linqProvier;
        Projector = projector;
    }

    protected virtual IQueryable<TDest> GetQueryable(TSpecification spec) => LinqProvider
        .GetQueryable<TSource>().ApplyIfPossible(spec).Project<TSource,
        TDest>(Projector).ApplyIfPossible(spec);

    public virtual IEnumerable<TDest> Ask(TSpecification spec) => GetQueryable(spec).ToArray();

    int IQuery<TSpecification, int>.Ask(TSpecification spec) => GetQueryable(spec).Count();
}

public class PagedQuery<TSortKey, TSpec, TEntity, TDto>: ProjectionQuery<TSpec, TEntity, TDto>,
IQuery<TSpec, IPagedEnumerable<TDto>>
    where TEntity : class, IHasId
    where TDto : class, IHasId
    where TSpec : IPaging<TDto, TSortKey>
{
    public PagedQuery(ILinqProvider linqProvier, IProjector projector) : base(linqProvier, projector) { }

    public override IEnumerable<TDto> Ask(TSpec spec) =>
        GetQueryable(spec).Paginate(spec).ToArray();

    IPagedEnumerable<TDto> IQuery<TSpec, IPagedEnumerable<TDto>>.Ask(TSpec spec)
        => GetQueryable(spec).ToPagedEnumerable(spec);

    public IQuery<TSpec, IPagedEnumerable<TDto>> AsPaged()
        => this as IQuery<TSpec, IPagedEnumerable<TDto>>;
}
```

Метод ApplyIfPossible проверит осуществляется фильтрация на уровне агрегата или проекции (бывает нужно и так, и так). Метод Project создаст проекцию с помощью AutoMapper.

Принципы CQRS очень хорошо подходят для реализации по протоколу HTTP. Спецификация HTTP четко говорит GET-запросы должны возвращать данные с сервера. POST, PUT, PATCH – изменять состояние. Хорошим тоном в web-программировании считается перенаправление на GET после выполнения POST-операции, например, отправки формы. Таким образом: GET - это Запрос POST/PUT/PATCH/DELETE – это Команда.

Подобная архитектура может показаться «перегруженной». Действительно подобный подход накладывает определенные ограничения к квалификации разработчиков:

- требуется понимание паттернов программирования и хорошее знание платформы;
- первоначальные вложения в код инфраструктуры.

Преимущества:

- четкое отделение домена от инфраструктуры;
- минимизация объема кода в проекте, устранение дублирования кода, устранение циклических зависимостей, устранение рутины, использование соглашений, вместо избыточных конфигураций;
- регламентирование бизнес-логики и общей структуры проекта;
- абстракция от серверной инфраструктуры, поддержка горизонтального масштабирования.

Выводы. В результате исследования была изучена архитектурная модель построения бизнес-приложений SQRS, предложена базовая реализация модели, проанализированы преимущества и недостатки данной модели. Приведены примеры реализации задач проекции и постраничного вывода данных. Данная модель хорошо себя зарекомендовала на существующих проектах. Описанные в работе подходы к проектированию использованы при выполнении ряда проектов медицинской тематики (американские веб-сайты, специализирующиеся на поиске докторов, локаций госпиталей, сервисов: <https://www.bannerhealth.com>, <https://www.eehealth.org>, <https://blog.hap.org>), при этом продемонстрирована их эффективность в процессе разработки, гибкость к изменяющимся требованиям заказчика, скорость разработки, унификация архитектуры и терминологии предметной области.

1. Мартин Фаулер. Архитектура корпоративных программных приложений - М.: «Вильямс», 2012. - 544 с.
2. Мартин Р., Мартин М. Принципы, паттерны и методики гибкой разработки на языке C#. - Пер. с англ. - СПб.: Символ-Плюс, 2011. - 768 с.
3. CQRS Documents by Greg Young: https://cQRS.files.wordpress.com/2010/11/cQRS_documents.pdf
4. CQRS by Martin Fowler: <http://martinfowler.com/bliki/CQRS.html>
5. Dino Esposito. Cutting Edge - CQRS for the Common Application: <https://msdn.microsoft.com/en-us/magazine/mt147237.aspx>

УДК 519.683.8

Гринюк С.В., Мельник В.М., Равенець А.В.
Луцький національний технічний університет

БУЛЕВІ ОПЕРАЦІЇ НАД ПОЛІГОНАМИ ПРИ СТВОРЕННІ МАПИ ШЛЯХІВ У ТРИВИМІРНОМУ ПРОСТОРІ ДЛЯ КОМП'ЮТЕРНИХ ІГОР

Гринюк С.В., Мельник В.М., Равенець А.В. Булеві операції над полігонами при створенні мапи шляхів у тривимірному просторі для комп'ютерних ігор. Ігри жанру RTS (real time strategy) використовують 3d компонент Mesh для визначення областей призначених для переміщення рухомих ігрових одиниць (мапа шляхів). Зазвичай це статична мапа яка не змінюється з часом гри. В даній статті розглядається проблема динамічної модифікації мапи шляхів з плином ігрового процесу, що дозволяє гравцю змінювати мапа шляхів розміщуючи різні ігрові об'єкти як перешкоди або новий шлях. Код представлений на мові програмування C#.

Ключові слова: ігри, 3d, булеві операції, мапа шляхів, c#, стратегії.

Гринюк С.В., Мельник В.М., Равенець А.В. Булевые операции над полигонами при создании карты путей в трехмерном пространстве для компьютерных игр. Игры жанра RTS используют 3d компонент Mesh для определения областей предназначенных для перемещения подвижных игровых единиц. Обычно это статическая карта которая не меняется со временем игры. В данной статье рассматривается проблема динамической модификации карты путей с течением игрового процесса, что позволяет игроку изменять карту путей размещая различные игровые объекты как препятствия или новый путь. Код представлен на языке программирования C#.

Ключевые слова: игры, 3d, булевые операции, карта путей, c#, стратегии.

Hrynyuk S., Melnyk V., Ravenets A. Boolean operations between polygons for creating paths map in tree-dimensional space for computer games. RTS games uses 3d component Mesh for defining zones designed for moving mobile game objects. Often it is a static map which is not changing during the gameplay. In here we analyzing problem of dynamic modification of paths map during gameplay which allows player change paths map by placing different game objects as obstacles or new path. Code is written using C# programming language.

Keywords: games, 3d, boolean operations, paths map, c#, strategies.

Постановка проблеми. Жанр відеоігор, в якому запорукою досягнення перемоги є планування і стратегічне мислення називаються – стратегії. Основною характеристикою цього жанру є керування великою кількістю об'єктів як: солдати, військова техніка, фантастичні звірі і т.д. Для того щоб підконтрольний об'єкт зміг дійти до пункту призначення використовують різні алгоритми пошуку шляху. Одним із найвідоміших є алгоритм A*. Однак, для 3d середовищ є більш ефективний метод який називається Navigation Mesh. Основними перевагами цього алгоритму є те що він працює не тільки в одній площині та замість розподілу усієї мапи шляхів на клітини (Nodes) опуклі полігони цього компоненту виступають в ролі клітин, але це не означає що вони містять в собі лише один крок, вони є зонами в яких об'єкт може вільно пересуватись у різних площинах. Ми розглядаємо випадок коли гравець може вносити зміни у мапу. Будувати різні будівлі які містять зони по яким можна пересуватись або ж є перешкодами. Оскільки мапа представлена як 3d компонент Mesh нам потрібно реалізувати зміну шляху, а саме такі логічні операції як сума та різниця.

Аналіз досліджень. За основу взятий ігровий двигун Unity3d. 3d об'єкти у ньому представлені у вигляді масиву точок та масиву трикутників, які в свою чергу складаються з перелічених за годинниковою стрілкою трьох точок. Для виконання логічних операцій був взятий алгоритм для полігонів [1]. Щоб реалізувати алгоритм нам потрібно утворити зв'язні контури використовуючи точки та трикутники. Після цього потрібно реалізувати процес триангуляції щоб вихідний об'єкт був коректний для використання у 3d.

Мета. Метою статті є успішне виконання булевих операцій суми та різниці для 3d компоненту Mesh на основі алгоритму для полігонів у 2d вимірі.

Виклад основного матеріалу й обґрунтування отриманих результатів.

Утворення контуру. Для того щоб виконати операцію суми чи різниці, нам потрібно підготувати вхідні данні. Алгоритм опирається на полігон що являє собою список точок та список контурних ребер [1]. Тому оскільки список точок у нас уже є, потрібно використовуючи трикутники сформувати список контурних ребер (контур). Для початку підготуємо типи які ми надалі будемо використовувати. Основними будуть три типи:

1. Точка (Vertex)
2. Ребро (Edge)
3. Трикутник (Triangle)

Оскільки уже існуючий в Unity3d клас який представляє собою точку у 3d просторі `Vector3` є структурою, ми створимо свій тип який буде класом. Для точки нам потрібно зберігати об'єкт `Vector3`, для ребра його дві точки, для трикутника - три точки і три ребра. Після того як ми сформували типи нам потрібно сформувати ці всі елементи. Точки у нас є тому ми просто використовуємо цикл формуємо список з нашими об'єктами `Vertex`. для трикутників виконуємо аналогічну операцію. Для ребер, нам потрібно з усіх утворених нами трикутників, вибрати такі ребра які з'являються лише в одному трикутнику, оскільки внутрішні ребра є суміжними для трикутників. Дані для контура готові. Головним моментом є те що ребра контура полігона повинні бути перераховані в списку за годинниковою стрілкою. Але дірки також виступають контуром для полігона тільки напрямком ребер проти годинникової стрілки.

Знаходження перетину ребер у просторі. Після того як ми сформували данні для двох контурів над якими ми будемо виконувати операції суми та різниці нам потрібно знайти усі можливі перетини між ними. Для початку нам потрібно визначити чи перетинаються ребра. Ми будемо використовувати матриці повороту для спрощення операцій з 3d виміру до 2d. Для підтвердження їхнього перетину нам достатньо довести наступне для кожного ребра:

- Кінцеві точки одного з ребер лежать в різних півплощинах які утворюються в результаті поділу площини на дві іншим ребром.

Ми знаємо що векторний добуток двох векторів дає нам третій вектор, напрямком якого залежить від того, додатний чи від'ємний кут між першим та другим векторами, відповідну така операція антикоммутативна. А так як вектори будуть лежати в двох осях (**перед перевіркою перетину нам потрібно повернути ребра на вісь Z так щоб усі точки мали однакове значення по осі Z щоб ми змогли ігнорувати це значення для подальших операцій та виконувати їх уже не в 3d, а в 2d просторі. А після отримання бажаного результату ми повернемо його в зворотньому напрямку на ті ж самі кути. Даний процес буде описаний нижче**) векторний добуток буде мати не нульову компоненту тільки Z.

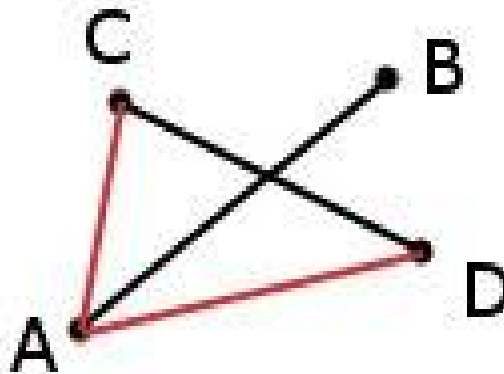


Рис. 1. Новоутворені вектори.

Відповідно відмінність векторних добутоків буде тільки в цій компоненті. Тому ми можемо помножити попарно-векторно вектор ребра що розділяє площину на півплощини і нові вектора направлені від початку розділяючого вектора до обох кінці вектора що перевіряється. Якщо компоненти Z обох добутоків буде мати різний знак, це означає що один з кутів менше 0, але не більше -180, а другий більше 0 і не більше 180 градусів. Відповідно точки лежать по різні сторони від прямої. Якщо ми маємо одну компоненту Z рівну нулю це означає що ребро дотикається іншого. Після цього на потрібно перевірити і інше ребро. Якщо ж два відрізки задовільняють усі умови то перетин існує.

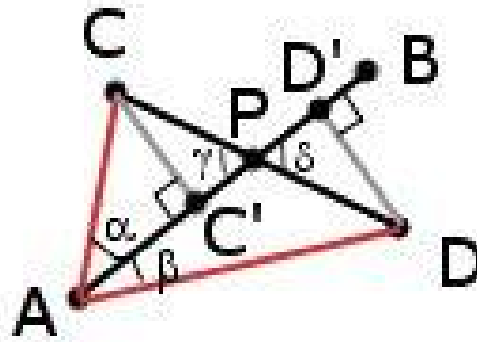


Рис 2. Висоти які є результатом скалярного добутку векторів.

Довжина векторного добутку векторів a і b (як ми з'ясували, що чисельно дорівнює його компоненті Z) дорівнює добутку модулів цих векторів на синус кута між ними.

$$|a| |b| \sin \alpha$$

Відповідно, для конфігурації на малюнку ми маємо наступне:

$$|AB \times AC| = |AB||AC| \sin \alpha, \text{ і } |AB \times AD| = |AB||AD| \sin \beta$$

$|AC| \sin \alpha$ є перпендикуляром, опущеним з точки C на відрізок AB , а $|AD| \sin \beta$ є перпендикуляром, опущеним з точки D на відрізок AB (катетом ADD'). Так як кути γ і δ - вертикальні кути, то вони рівні, а значить трикутники PCC' і PDD' подібні, а відповідно і довжини всіх їх сторін пропорційні в рівне ставлення.

Маючи $Z1$ ($AB \times AC$, а значить $|AB||AC| \sin \alpha$) і $Z2$ ($AB \times AD$, а значить $|AB||AD| \sin \beta$), ми можемо розрахувати $CC' + DD'$ (яка дорівнюватиме $Z1 + Z2$), а також знаючи що $CC' + DD' = CP + DP$ легко можна вирахувати місце розташування точки P . Ось кінцеві формули:

$$Px = Cx + (Dx - Cx) * |Z1| + |Z2 - Z1|$$

$$Py = Cy + (Dy - Cy) * |Z1| + |Z2 - Z1|$$

Вигляд методу на C#:

(Алгоритм виконується для попередньо повернутих ребер.)

```
public Vertex GetIntersection(Edge edge1, Edge edge2, ref bool bound)
{
    bool boundLocal = false;

    Vector Vector1 = edge1.GetVector();
    Vector Vector2 = edge2.GetVector();

    Vector Prod1 = CrossProduct(Vector1, new Edge(
        edge1.Start,
        edge2.Start
    ).GetVector());

    Vector Prod2 = CrossProduct(Vector1, new Edge(
        edge1.Start,
        edge2.End
    ).GetVector());
```

```
if (!boundLocal && (Prod1.y == 0f || Prod2.y == 0f))
{
    boundLocal = true;
}

// Граничні значення рахуються перетином
if ((Prod1.y == 0f && Prod2.y == 0f) || Math.Sign(Prod1.y) == Math.Sign(Prod2.y))
{
    return null;
}

Prod1 = CrossProduct(Vector2, new Edge(
    edge2.Start,
    edge1.Start
).GetVector());

Prod2 = CrossProduct(Vector2, new Edge(
    edge2.Start,
    edge1.End
).GetVector());

if (!boundLocal && (Prod1.y == 0f || Prod2.y == 0f))
{
    boundLocal = true;
}

// Граничні значення рахуються перетином
if ((Prod1.y == 0f && Prod2.y == 0f) || Math.Sign(Prod1.y) == Math.Sign(Prod2.y))
{
    return null;
}

// Опрацювання ділення на нуль.
if (Prod2.y == Prod1.y)
{
    return null;
}

float x = edge1.Start.x + Vector1.x * Math.Abs(Prod1.y) / Math.Abs(Prod2.y - Prod1.y);
float z = edge1.Start.z + Vector1.z * Math.Abs(Prod1.y) / Math.Abs(Prod2.y - Prod1.y);

bound = boundLocal;

return new Vertex(x, 0f, z);
}

public Vertex GetVector()
{
    return new Vertex(
        End.x - Start.x,
        End.y - Start.y,
        End.z - Start.z
    );
}

public Vertex CrossProduct(Vertex Vector1, Vertex Vector2)
{
    return new Vertex(
        (Vector1.y * Vector2.z - Vector1.z * Vector2.y),
        (Vector1.z * Vector2.x - Vector1.x * Vector2.z),
        (Vector1.x * Vector2.y - Vector1.y * Vector2.x)
    );
}
```

Поворот точок в просторі. Перед тим як виконувати поворот точки нам потрібно визначити

нормалію площини в якій лежать наш контур, що являє собою току в просторі яка належить площині і через яку проходить перпендикуляр до цієї площини від початку координат (0, 0, 0). Ми не будемо зупинятися на цьому кроці, в мережі доступно багато способів визначення цієї точки. Коли ми отримали нормалію площини ми можемо визначити кут для повороту площини спочатку по осі Y, а потім по осі X. Для першого випадку це буде косинус кута між проекцією перпендикуляра на осі X-Z і відрізком що лежить на осі Z. Для другого випадку косинус кута між перпендикуляром та відрізком що лежить на осі Y. Після того як ми за допомогою векторів знайшли косинуси кутів, нам потрібно дізнатися їх синус, але з правильним знаком. Синус ми знайдемо легко за допомогою формули:

$$\cos \alpha + \sin \alpha = 1$$

А знак ми дізнаємось з того в якій четверті лежать проекція перпендикуляра на осі X-Z відносно цих осей і проекція перпендикуляра на осі Y-Z відносно цих осей для знайдених косинусів відповідно.

Якщо проекції лежать на одній прямій з відповідними осями до яких визначається кут або лежать в четвертях 2 і 3 тоді напрямком за годинниковою стрілкою, тобто знак плюс. Якщо ж вони лежать в четвертях 1 і 4 тоді напрямком проти годинникової стрілки тобто знак мінус. На даному етапі ми знайшовши два кути по яких ми будемо виконувати поворот точок площини. Данні значення варто обчислити один раз і зафіксувати для усіх компонентів що знаходяться в тій самій площині. Значення синусів і косинусів використовуються у формулах матриць повороту.

Маркування ребер. Перед тим як розпочати маркування ребер, нам потрібно розбити ребра які містять в собі точки перетину для двох контурів. Далі слід маркувати ребра, для цього ми будемо використовувати два показника для кожного ребра fl і fr. Для усіх ребер з першого контуру присвоюємо значення fl = 1, а fr = 0. Для усіх ребер з другого контуру які ще не марковані присвоюємо значення fl = 2, а fr = 0. Якщо ребро з другого контуру існує в першому у нас є два варіанта. Якщо їх напрямок співпадає (напрямок можна перевірити по точкам ребер, start і end) то ми збільшуємо для нього fl на 2 (отримуємо fl = 3 fr = 0), якщо напрямок різний присвоюємо fr = 2 (отримуємо fl = 1 fr = 2). Після цього нам потрібно опрацювати ребра які знаходяться всередині контурів. Для цього ми використаємо алгоритм належності точки полігону (**описаний далі**). Для всіх ребер з першого контуру перевіряємо чи існує таке ж ребро не зважаючи на напрямок в другому контурі. Якщо ні знаходимо центральну точку ребра і перевіряємо чи належить вона другому контуру. Якщо вона належить збільшуємо показники fl і fr ребра на 2 (отримаємо fl = 3 fr = 2. було fl = 1 fr = 0). Теж саме виконуємо аналогічно для ребер другого контуру тільки якщо умови виконуються збільшуємо показники не на 2 а на 1 (отримаємо fl = 3 fr = 1. було fl = 2 fr = 0). На даному етапі усі ребра марковані нам залишилось тільки вибрати ті які потрібні за допомогою таблиці:

Таблиця 1. Вибірка ребер

fl	fr	Сума	Переріз	Різниця	Симетрична різниця
1	0	+	-	+	+
2	0	+	-	-	+
1	2	-	-	+	-
3	0	+	+	-	-
3	1	-	+	<>	<>
3	2	-	+	-	<>

Позначення в таблиці:

+ додаємо ребро до нового контура

- не додаємо ребро до нового контура

<> додаємо ребро до нового контура, але змінюємо його напрям

Опираючись на таблицю утворюємо нові контури визначаємо чи виступає якийсь контур діркою, та формуємо новий полігон. Ось і все, на даному етапі маємо новий полігон що є результатом булевої операції над двома іншими полігонами.

1. Леонов М. В., Никитин А. Г. Эффективный алгоритм, реализующий замкнутый набор булевых операций над множествами многоугольников на плоскости / Препринт Института систем информатики СО РАН № 46. 1997. — 20 с.
2. Скворцов А.В. Построение объединения, пересечения и разности произвольных многоугольников в среднем за линейное время с помощью триангуляции // Вычислительные методы и программирование, 2002, Т. 3. С. 116-123.
3. Скворцов А. В. Линейно-узловой алгоритм построения оверлеев двух полигонов // Вестник ТГУ. 2002. № 275. С. 99—103.
4. Ченцов О.В., Скворцов А.В. Обзор алгоритмов построения оверлеев многоугольников // Вестник ТГУ. 2003. № 280. С. 338–345.

УДК 004.415.2

Гринюк С.В., Поліщук М.М., Грабовський Б.М.
Луцький національний технічний університет

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ЗБЕРІГАННЯ І ОПЕРУВАННЯ ДАНИМИ ПРО НАДАНІ ПОСЛУГИ І ПРОДАНІ ТОВАРИ ЗАСОБАМИ C++ ТА QT

Гринюк С.В., Поліщук М.М., Грабовський Б.М. Проектування та розробка програмної системи для зберігання і оперування даними про надані послуги і продані товари засобами C++ та QT. В статті розглядається проектування та створення програми-системи обліку проданих товарів та наданих послуг. Функціональність системи розподіляється між двома користувачами, але тільки адміністратор має можливості додавання і редагування товарів, налаштування для роботи з зовнішнім сервісом та додавання працівників з можливістю зміни про них інформації.

Ключові слова: адміністратор, програмне забезпечення, сервер.

Гринюк С.В., Поліщук М.М., Грабовський Б.М. Проектирование и разработка программной системы для хранения и оперирования данными о предоставленных услугах и проданных товарах средствами C++ и QT. В статье рассматривается проектирование и создание программы-системы учета проданных товаров и оказанных услуг. Функциональность системы распределяется между двумя пользователями, но только администратор имеет возможности добавления и редактирования товаров, настройки для работы с внешним сервисом и добавления работников с возможностью изменения о них информации.

Ключевые слова: администратор, программное обеспечение, сервер.

Hrynyuk S., Polishchuk M., Hrabovskiy B. Design and development of a software system for storing and handling data on services rendered and goods sold means of C++ and QT. In the article the design and creation of program accounting system of goods sold and services rendered. System functionality shared between two users, but the administrator has the possibility of adding and editing products, setting to work with external service and adding workers with the ability to change them information.

Keywords: Administrator, program Provision server.

Постановка проблеми. Основною задачею є проектування та створення програми-системи обліку проданих товарів та наданих послуг для магазину.

Щоб реалізувати дану систему перш за все потрібно виділити основні дії, які вона повинна виконувати., а саме:

- зберігання інформації про клієнтів;
- створення замовлень;
- розподіл організаційних функцій між працівниками;
- взаємодія з товарами та послугами.

Все це лише основні вимоги до програми. Насправді таких вимог набагато більше. Вони будуть формуватися і, можливо, модифікуватися в процесі розробки.

Аналіз останніх досліджень і публікацій. Під час огляду аналогів було виділено декілька основних функцій, що має містити у собі кожна програма:

- заносити наявні товари в систему;
- змінювати інформацію про товари;
- додавати працівників;
- редагувати працівників;
- створювати замовлення для товарів і послуг;
- друкувати чеки;
- сповіщати замовників про виконання послуги з допомогою повідомлень на мобільний телефон.

Метою даної статті є проектування та розробка програмної системи для зберігання і оперування даними про надані послуги і продані товари.

Основна частина

Автоматизована інформаційна система – це пов'язана сукупність даних, програмного забезпечення, апаратного забезпечення, персоналу, функцій та процедур, які виконують пошуку, збір, обробку, розподіл, зберігання, представлення інформації у відповідності з вимогами, які виходять із цілей організації. На сьогоднішній день кожна інформаційна система використовує комп'ютерні технології.

Головне завдання інформаційної системи управління – підпорядкування всіх внутрішніх процесів головним цілям організації. Для цього потрібно організувати процеси, які пов'язані з діяльністю організації так, щоб вони могли максимально забезпечити виконання поставлених завдань в єдиному інформаційному полі.

Для розробки системи було вибрано мову програмування C++, оскільки вона є однією з найшвидших та фреймворк QT [5], який надає засоби для створення інтерфейсу, а також значно полегшує і пришвидшує розробку забезпечуючи додаткові засоби у вигляді модулів які підключаються у разі необхідності для реалізації потрібної функціональності.

Програмна система буде розділяти функціональність між двома користувачами: адміністратором та продавцем [2]. Діаграма варіантів використання для адміністратора зображена на рисунку 1.

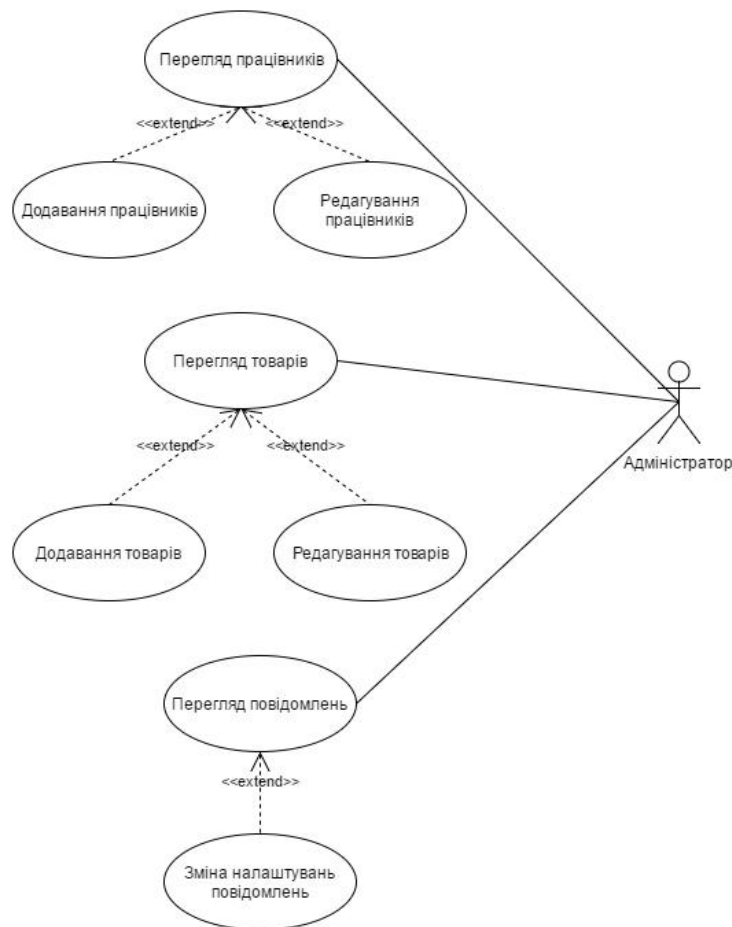


Рис. 1. Діаграма варіантів використання для користувача «Адміністратор»
Таблиця 1. Опис прецедентів для користувача «Адміністратор»

Прецедент	Короткий опис
Перегляд працівників	Дозволяє переглядати працівників системи.
Додавання працівників	Дозволяє створювати нових працівників.
Редагування працівників	Дозволяє редагувати інформацію про працівників.
Перегляд товарів	Дозволяє переглядати всі зареєстровані товари.
Додавання товарів	Дозволяє додавати нові товари.
Перегляд повідомлень	Дозволяє переглядати відправлені повідомлення та їх статуси.
Зміна налаштувань повідомлень	Дозволяє задавати та змінювати налаштування, необхідні для відправлення повідомлень.

Основна функціональність роботи з замовленнями послуг і товарів покладається на користувача «Продавець», діаграма варіантів використання якого зображена на рисунку 2.

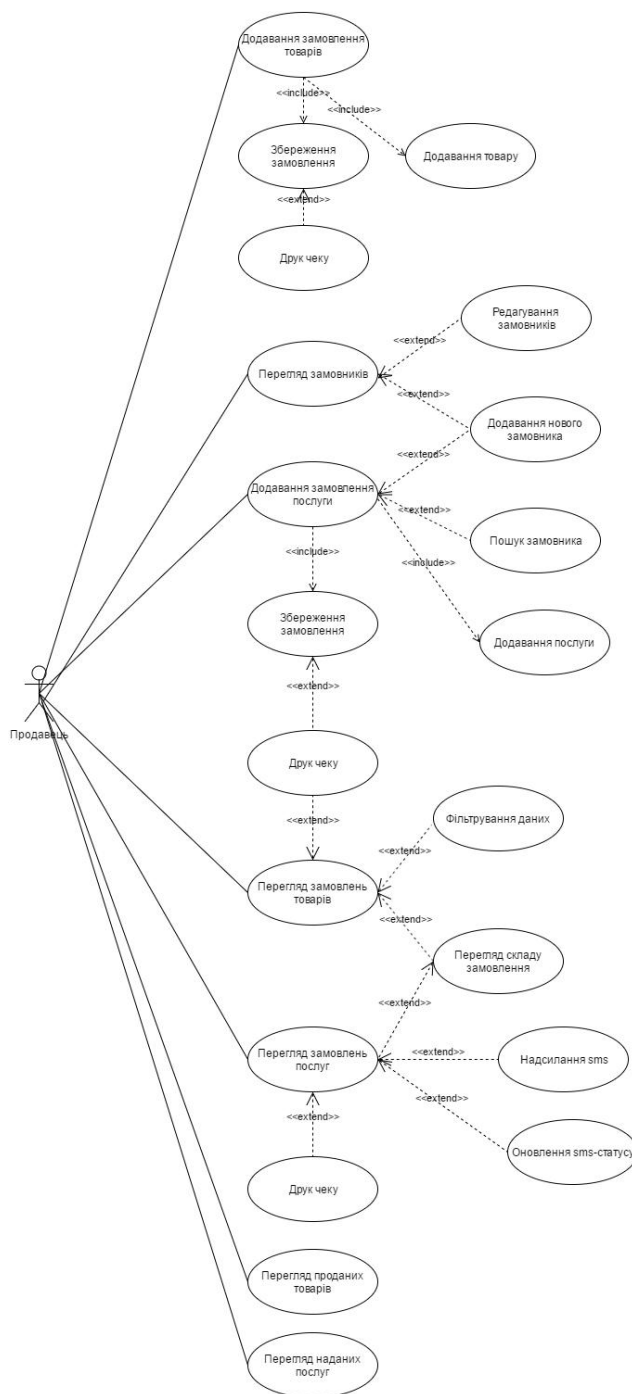


Рис. 2. Діаграма варіантів використання для користувача «Продавець»

Приклад функції для відправлення запиту надсилання повідомлення до зовнішнього сервісу (лістинг 1) [1].

Лістинг 1 – Функція відправлення запиту надсилання повідомлення до зовнішнього сервісу

```

void SmsClubService::send( QString receiverNumber, QString message )
{
    // Отримуємо повідомлення в необхідному кодуванні
    QByteArray encodedString = getEncodedMessage( message );
    QString emsg = encodedString.toBase64();
}

```

```
// Читаємо налаштування для відправлення повідомлення
auto settings = _setting->readSetting();
// Створюємо запит до зовнішнього сервісу
QNetworkRequest request( _requests[ Requests::SEND_MESSAGE ]
    .arg( settings.login )
    .arg( settings.password )
    .arg( settings.alfaName )
    .arg( receiverNumber )
    .arg( emsg ) );

// Встановлюємо конфігурацію для протоколу SSL
setSslConfiguration( request );
// Створюємо об'єкт через надсилання запиту, який відповідає за
// відповідь з сервісу
createNetworkReply( request );

// З'єднуємо сигнал об'єкта відповіді з функцією-обробником успішної
// відповіді
connect( _networkReply.data()
    , SIGNAL( readyRead() )
    , SLOT( readyReadSendMessage() ) );

// З'єднуємо сигнал об'єкта відповіді з функцією-обробником помилки
connect( _networkReply.data()
    , SIGNAL( error( QNetworkReply::NetworkError ) )
    , SLOT( replyError( QNetworkReply::NetworkError ) ) );
}
```

Для роботи нашої програмної системи необхідно спочатку її розгорнути. Оскільки програма є клієнт-серверною, для її встановлення знадобиться як мінімум два комп'ютери, один для сервера і один для клієнта [3]. Діаграма розгортання зображена на рисунку 3.

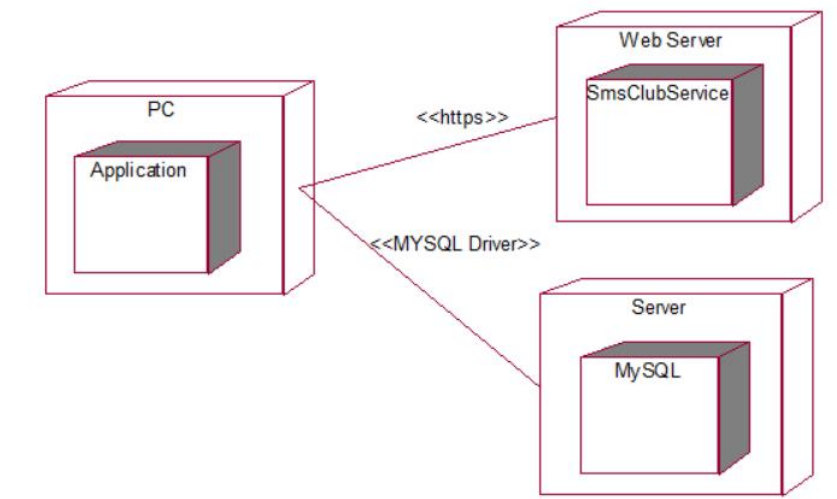


Рис. 3. Діаграма розгортання програмної системи

Як видно з рисунку 3, програма на клієнті взаємодіє з сервером бази даних за допомогою MySQL драйвера, який постачається разом з системними бібліотеками QT [6]. Також клієнту необхідний доступ до мережі Інтернет, щоб з'єднатися з зовнішнім веб-сервером з допомогою протоколу https, який надасть можливість відправляти повідомлення.

Системні вимоги до сервера бази даних:

- процесор 2.0 ГГц;
- оперативна пам'ять 2 ГБ;
- об'єм жорсткого диску 200 ГБ.

Системні вимоги до клієнта:

- процесор 2.0 ГГц;
- вільна оперативна пам'ять 256 Мб;
- необхідний вільний простір на жорсткому диску – 100 Мб.

Перед початком роботи з програмною системою необхідно зареєструватися на сайті <https://smsclub.mobi/> і задати клієнту відповідні налаштування з панелі адміністратора.

Перед початком роботи необхідно авторизуватися в системі. Форма входу зображена на рисунку 4.

Рис. 4. Форма входу

Форма входу дозволяє авторизуватися в системі вводячи логін і пароль. Для того, щоб зайти з правами адміністратора необхідно вибрати пункт «Зайти як адміністратор». В іншому випадку вхід буде здійснюватися з правами продавця.

Висновки. Результатом виконання роботи є програмна система для обліку проданих товарів та наданих послуг.

Під час створення програмного продукту було проаналізовано предметну область, виділено акторів системи та їх варіанти використання, спроектовано базу даних. Також розроблено архітектуру програми з реалізацією всієї необхідної функціональності, що була поставлена в завданні. Програмна система успішно пройшла верифікацію. Для забезпечення коректної роботи були сплановані і проведені тести.

Таким чином, розроблена програмна система реалізовує для адміністратора можливості додавання і редагування товарів, налаштування для роботи з зовнішнім сервісом та додавання працівників з можливістю зміни про них інформації.

- 1.Макконел С. Совершенный код. Мастер-класс / Пер. с англ. [Текст] – М. : Издательство «Русская редакция», 2010. – 896 с.
- 2.Пасічник В. В., Резніченко В.А. - Організація баз даних та знань. [Текст] – К.: Видавнича група ВНУ, 2013. – 384 с.
- 3.Скотт Мейерс Эффективный и современный C++: 42 рекомендации по использованию C++11 и C++14. [Текст]: Пер. с англ. - М.: ООО "ИЛ. Вильямс", 2016. - 304 с.: ил. - Парал. тит. англ.
- 4.C++ Стандартная библиотека. Для профессионалов / Н. Джосьютис. [Текст] – СПб: Питер, 2012. – 730 с.
- 5.Шлее М. QT 5.3. Профессиональное программирование на C++ [Текст] – СПб.: БХВ-Петербург, 2015. – 928 с.: ил. – (В подлиннике).
- 6.MySQL 5.0. Библиотека программиста / Виктор Гольцман [Текст] – Питер, 2010. – 253 с.
- 7.Алгоритмы. Построение и анализ. Третье издание / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. [Текст] – Пер. с англ. – М. : Издательский дом "Вильямс", 2013. – 1428 с. : ил. – парал. тит. англ.

УДК 004.434

Г. М. Губаль

Луцький національний технічний університет

СИСТЕМА L^AT_EX У СТВОРЕННІ КОМП'ЮТЕРНИХ МАТЕМАТИЧНИХ ТЕКСТІВ

Губаль Г.М. Система L^AT_EX у створенні комп'ютерних математичних текстів. У статті наведено і проаналізовано деякі правила створення багаторядкових внутрішньотекстових і виключних математичних формул, позначення компонент і типів символів засобами L^AT_EX. Досліджено деякі можливості пакету amsmath, які є актуальними при створенні математичних текстів.

Ключові слова: система L^AT_EX, багаторядкова формула, \vphantom, \bmatrix, \declaremathoperator.

Літ. 15.

Губаль Г.Н. Система L^AT_EX в создании компьютерных математических текстов. В статье приведено и проанализировано некоторые правила создания многострочных внутритекстовых и выключных математических формул, обозначения компонент и типов символов средствами L^AT_EX. Исследовано некоторые возможности пакета amsmath, актуальные при создании математических текстов.

Ключевые слова: система L^AT_EX, многострочная формула, \vphantom, \bmatrix, \declaremathoperator.

Лит. 15.

Hubal H.M. The system L^AT_EX in the creation of computer mathematical texts. Some rules for creation of multi-line inline and display mathematical formulae, component notations and character types notations are given and analyzed by means of L^AT_EX in the paper. Some opportunities of the amsmath package are investigated, relevant when generating mathematical texts.

Keywords: system L^AT_EX, multi-line formula, \vphantom, \bmatrix, \declaremathoperator.

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 правила створення багаторядкових внутрішньотекстових і виключних математичних формул, позначення компонент і типів символів засобами L^AT_EX, а також досліджено деякі можливості пакету amsmath, які є актуальними при створенні математичних текстів.

Основна частина. Розглянемо створення багаторядкових внутрішньотекстових і виключних математичних формул у математичних текстах засобами L^AT_EX. Загальним правилом є наведення знаків „бінарного відношення” або „бінарної операції” в кінці рядка внутрішньотекстової формули або на початку і в кінці рядка виключної формули.

У математичних текстах внутрішньотекстові формули [1, 10] генеруються командою $\$...\$$ або командою $\backslash(...\backslash)$. Частини довгої внутрішньотекстової формули автоматично переносяться L^AT_EX на інший рядок (наприклад, після знаків „бінарного відношення” або „бінарної операції”). Якщо необхідно уникнути автоматичних переносів у частині внутрішньотекстової формули, то беремо її у фігурні дужки. При цьому ускладнюється верстка абзаців, внаслідок чого можливі переповнення у рядках. Щоб при верстці абзаців було менше переповнень, у преамбулі документа слід записати такі оператори:

```
\binoppenalty=10000  
\relpenalty=10000
```

Ці оператори визначають ступінь небажаності розриву рядка після знаків „бінарної операції” і „бінарного відношення” відповідно.

У математичних текстах виключні формули генеруються командою $\backslash[...]$ або оточенням equation пакета amsmath. Довгі виключні формули автор розбиває самостійно на кілька рядків за допомогою таких оточень пакета amsmath, як multline, align, aligned, alignat, split, gather.

Розглянемо механізм автоматичного вибору розміру дужок у багаторядковій виключній формулі засобами L^AT_EX. Команди \left і \right дозволяють автоматично вибирати розміри відкриваючої і закриваючої дужок відповідно. При цьому перед переходом на наступний рядок необхідно закрити команду \left або \right відповідною командою \right. або \left. Наприклад, код

```
\begin{multline*}
\int\limits_{\{\mathbb{R}^{\nu}\}} \times \{\mathbb{R}^{\nu}\}
\{\mathrm{d}\{x_2\} \setminus \{H_2\}, \setminus \{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\} \} =
\int\limits_{\{\mathbb{R}^{\nu}\}} \times \{\mathbb{R}^{\nu}\}
\{\mathrm{d}\{x_2\}
\left\{ \left\{ \sum\limits_{i=1}^2 \frac{p_i^2}{2} + \Phi(\{q_1\} - \{q_2\}) \right\}, \right\} \setminus
\{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\} \right\} \setminus
\end{multline*}
```

не може бути скопійований L^AT_EX, оскільки усі команди \left повинні бути закриті відповідними командами \right у кожному рядку багаторядкової виключної формули. Тому у кожному рядку необхідно команди \left і \right закривати командами \right. і \left. відповідно. Наприклад, код

```
\begin{multline*}
\int\limits_{\{\mathbb{R}^{\nu}\}} \times \{\mathbb{R}^{\nu}\}
\{\mathrm{d}\{x_2\} \setminus \{H_2\}, \setminus \{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\} \} =
\int\limits_{\{\mathbb{R}^{\nu}\}} \times \{\mathbb{R}^{\nu}\}
\{\mathrm{d}\{x_2\}
\left\{ \left\{ \sum\limits_{i=1}^2 \frac{p_i^2}{2} + \Phi(\{q_1\} - \{q_2\}) \right\}, \right\} \right. \setminus
\left. \{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\} \right\} \setminus
\end{multline*}
```

генерує таку формулу:

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

Як бачимо, розмір дужок для кожної пари команд \left і \right генерується незалежно. У цьому випадку необхідно знайти найвищий елемент у формулі. Для наведеного прикладу найвищим елементом є $\sum_{i=1}^2 \frac{p_i^2}{2}$. Використаємо екзотичну команду вертикального фантома \vphantom, яка шукає найвищий елемент. Тоді код

```
\begin{multline*}
\int\limits_{\{\mathbb{R}^{\nu}\}} \times \{\mathbb{R}^{\nu}\}
\{\mathrm{d}\{x_2\} \setminus \{H_2\}, \setminus \{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\} \} =
\int\limits_{\{\mathbb{R}^{\nu}\}} \times \{\mathbb{R}^{\nu}\}
\{\mathrm{d}\{x_2\}
\left\{ \left\{ \sum\limits_{i=1}^2 \frac{p_i^2}{2} + \Phi(\{q_1\} - \{q_2\}) \right\}, \right\} \right. \setminus
\left. \{F_2(t, \{x_1\}, \{x_2\} | F_1(t))\} \right\}
\vphantom{\sum\limits_{i=1}^2 \frac{p_i^2}{2}} \right\} \setminus
\end{multline*}
```

генерує таку формулу:

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

Іншим способом вирішення цієї проблеми є використання фіксованої висоти дужок замість пошуку найвищого елемента у формулі. Наприклад, код

```
. \begin{multline*}
\int\limits_{\{\mathbb{R}^{\nu}\}} \times \{\mathbb{R}^{\nu}\}
```

```
\differential{d}{x_2}\{ {H_2},\{F_2(t,{x_1},{x_2})|{F_1(t)}\} \}=
\int\limits_{\{\mathbb{R}\}^{\nu}}\{\mathbb{R}\}^{\nu}}
\differential{d}{x_2}
\Bigl\{\sum\limits_{i=1}^2\{\frac{p_i^2}{2}+\Phi(q_1-q_2)\},\}\Bigr\}\Bigr\}
\{F_2(t,{x_1},{x_2})|{F_1(t)}\}\Bigr\}.
```

генерує таку формулу:

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

Зауважимо, що цей спосіб застосовний лише у випадках, в яких дужки не вищі від \Bigg.

Автоматичне вирішення цієї проблеми є використання пакету breqn, розробленого М. Доуномсом і М. Хогольмом, у преамбулі документа. При цьому команди \left і \right працюють правильно і у випадку, коли формула є багаторядковою.

У деяких випадках необхідно явно показати матричні і векторні компоненти. У L^AT_EX матриці і вектори необхідно записувати за допомогою таких оточень пакета amsmath: pmatrix (для круглих дужок) і bmatrix (для квадратних дужок). Наприклад, код

```
\begin{bmatrix}
a_{11}&a_{12}&a_{13}\\
a_{21}&a_{22}&a_{23}\\
a_{31}&a_{32}&a_{33}
\end{bmatrix}
\begin{bmatrix}
x_1\\x_2\\x_3
\end{bmatrix}=
\begin{bmatrix}
b_1\\b_2\\b_3
\end{bmatrix}
```

генерує такий результат:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Розглянемо типи символів у математичному режимі L^AT_EX. У математичному режимі символи друкуються курсивом за замовчуванням. Наприклад, код

```
\[x=y]
```

генерує такий результат:

$$x = y$$

Щоб змінити шрифт деяких символів на прямий у математичному режимі, використовується команда \mathrm. Наприклад, код

```
\[ \mathrm{x} = \mathrm{y} ]
```

генерує такий результат:

$$x = y$$

Зауважимо, що ця команда ігнорує пробіли. Якщо необхідно створити пробіли, використовують команду \text. Наприклад, код

$\mathrm{a\,b}$
 $\text{a\,b}$

генерує такий результат:

$$\begin{matrix} S_{ab} \\ S_{a\,b} \end{matrix}$$

Зауважимо, що команда `\text` може задавати курсивний шрифт. Наприклад, код

`\textit{Оператори Гамільтона і Ліувілля позначають}\(\mathrm{H}_s\),  $H_s$ \textit{відповідно.}}`

генерує такий результат:

Оператори Гамільтона і Ліувілля позначають H_s , H_s відповідно.

Пакет `amsmath` дає можливість створювати свої математичні оператори у преамбулі документа за допомогою команди `\DeclareMathOperator` після `\usepackage{amsmath}` і в тексті за допомогою команди `\operatorname`, які генерують прямий шрифт для назви математичного оператора та формують пробіл між цим оператором і його аргументом. Наприклад, команда

`\DeclareMathOperator{\Re}{Re}`

у преамбулі документа і команда

`\Re z`

генерує формулу

$\operatorname{Re} z$,

яку генерує і команда

`\operatorname{Re}z`

у тексті документа.

Висновки. Таким чином, у цій статті наведено і проаналізовано деякі правила створення багаторядкових внутрішньотекстових і виключних математичних формул, позначення компонент і типів символів засобами $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$, а також досліджено деякі можливості пакету `amsmath`, які є актуальними при створенні математичних текстів.

1. Балдин Е. М. Компьютерная типография $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ / Е. М. Балдин. – СПб.: БХВ-Петербург, 2008.
2. Беляков Н. С. $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ для всех / Н. С. Беляков, В. Е. Палощ, П. А. Садовский. – М.: Книжный дом «Либроком», 2009.
3. Губаль Г. М. $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ як видавничча 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 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ 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 $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}2_{\epsilon}$ / М. Ю. Жуков, Е. В. Ширяева. – Ростов н/Д: Изд-во ООО «ЦВВР», 2003.
6. Жуков М. Ю. $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}2_{\epsilon}$: 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 $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ / Д. Е. Кнут. Пер. с англ. М. В. Лисиной. – Протвино: АО $\mathrm{RDT}_{\mathrm{E}}\mathrm{X}$, 1993.
9. Котельников И. А. $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ по-русски / И. А. Котельников, П. З. Чеботаев. – Новосибирск: Сибирский хронограф, 2004.
10. Львовский С. М. Набор и вёрстка в системе $\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}$ / С. М. Львовский. – 3-е изд., испр. и доп. – М.: МЦНМО, 2003.
11. Ширяева Е. В. Введение в $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ -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 378.146

Гулай О.І., д.п.н.

Луцький національний технічний університет

МЕТОДИКА ОЦІНЮВАННЯ НАВЧАЛЬНИХ ДОСЯГНЕНЬ СТУДЕНТІВ

Гулай О.І. Методика оцінювання навчальних досягнень студентів. У статті окреслюються завдання, вимоги, що висуваються до контролю, його функції, види, форми та методи. Метою є розроблення методики контролю при вивченні природничонаукових дисциплін. Інноваційні методи оцінювання були інтегровані в процес викладання і навчання. Методика зменшує упереджений підхід до оцінювання знань студентів. Вона є ефективним засобом мотивації навчально-пізнавальної діяльності та підвищує відповідальність студента за результати навчання.

Ключові слова: контроль, види і форми контролю, методи контролю.

Гулай О.И. Методика оценивания учебных достижений студентов. В статье описываются задачи, требования, предъявляемые к контролю, его функции, виды, формы и методы. Целью является разработка методики контроля при изучении естественнонаучных дисциплин. Инновационные методы оценки были интегрированы в процесс преподавания и обучения. Методика уменьшает предубежденный подход к оцениванию знаний студентов. Она является эффективным средством мотивации учебно-познавательной деятельности и повышает ответственность студента за результаты обучения.

Ключевые слова: контроль, виды и формы контроля, методы контроля.

Hulai O.I. Methods of evaluating educational achievements of students. The paper outlines the objectives and requirements that apply to control its functions, types, forms and methods. The aim is to develop control methods in the study of natural science disciplines subjects. Innovative assessment practices were integrated into the teaching and learning process. The methodology reduces the prejudiced approach to assessing students' knowledge. It is an effective tool for motivating educational and cognitive activities and increases student responsibility for learning outcomes.

Keywords: control, types and forms of control, methods of control.

Постановка наукової проблеми. Необхідність перебудови структури і змісту підготовки майбутніх фахівців, зумовлена потребою суспільства у висококваліфікованих кадрах, ставить нові вимоги до моніторингу освіти. Сучасна парадигма освіти змінюється від знаннєвої до компетентнісної, відповідно змінюються і технології оцінювання навчальних досягнень студентів. За останні роки значного розвитку набуває дистанційна модель навчання з системою управління знаннями, яка включає: управління зовнішніми потоками інформації; навчання; обмін знаннями; структурування знань в навчальному закладі; спільну роботу в групах; управління взаємовідносинами з представниками інших освітніх систем тощо [3]. Розвиток різноманітних видів педагогічного контролю стимулює навчання та пізнавальну діяльність студентів. При формуванні якості продукту освітнього проекту процес моніторингу повинен забезпечити постійне відстежування, аналіз та звітування про компетенції, які поступово формуються у студента протягом навчання в коледжі чи університеті [7]. Тому актуальною є проблема практичної реалізації методик оцінювання навчальних досягнень студентів із природничонаукових та фахових дисциплін.

Аналіз досліджень. Теоретичні проблеми контролю знань студентів були предметом досліджень В. Аванесова [1], В. Беспалька [2], Ю. Романенка [8], Л. Романишиної [9], Н. Тверезовської [11] та ін., технології діагностики та оцінювання навчальних досягнень – Т. Лукіної [5], Смолінчук Л. [10], теорія і практика створення тестів – М. Челишкової [12], В. Шахова [13], особливості комп'ютерного тестування розглянуті у праці Ю. Дорошенка, П. Ротаєнка [4] та ін.. Впровадження компетентнісного підходу зумовлює використання нових методів і прийомів поточного та екзаменаційного контролю у практиці вищої школи.

Компетентнісна парадигма навчання передбачає використання тестових технологій як індикатора успішності формування знаннєвої складової компетентностей різних рівнів. Світова тенденція розробки компетентнісних тестів спрямована не тільки на оцінювання результатів навчання, а й на проектування самого процесу навчання. Тестування у змозі вирішити одне з найскладніших завдань освіти – сформувати потребу і вміння самоаналізу і самоконтролю, що є фундаментом індивідуалізації та гуманізації освіти, орієнтування студентів на самоосвіту [10]. До переваг тестових технологій відносимо наступні чинники:

- велика об'єктивність і, як наслідок, позитивний стимулювальний вплив на пізнавальну діяльність студента;
- зниження негативного впливу на результати тестування таких суб'єктивних чинників, як настрої, рівень кваліфікації, особисті вподобання та інші характеристики конкретного викладача;
- орієнтованість на сучасні технічні засоби, зокрема комп'ютерні технології;

– універсальність, охоплення всіх стадій процесу навчання.

Погоджуємося з твердженнями О. Ляшенка та С. Ракова [6], що особливе місце тестових технологій обумовлено їхньою об'єктивністю (мінімізацією суб'єктивного фактора) та технологічністю (використання формалізованих процедур перевірки їх виконання). Однак, для встановлення рівня сформованості діяльнісного компонента компетентності тестові методики є малоефективними, тому нами застосовувалися методи оцінки результатів діяльності (виконання лабораторних та практичних робіт, розв'язання професійно спрямованих задач, індивідуальних розрахункових та дослідницьких завдань, захист проектів будівельного спрямування тощо). У європейських ВНЗ практикується також і залучення студентів до розроблення оцінювальних методик [14].

Мета дослідження – розроблення та апробація методики педагогічного контролю, що інтегративно поєднує різнопланові способи оцінювання навчальних досягнень студентів.

Виклад основного матеріалу і обґрунтування отриманих результатів. При розробленні та впровадженні методики педагогічного контролю навчально-пізнавальної діяльності студентів наші зусилля були спрямовані на вирішення наступних завдань:

- визначення якості засвоєння теоретичного навчального матеріалу, ступеня відповідності отриманих практичних умінь і навичок цілям і завданням навчальної дисципліни;
- виявлення труднощів у застосуванні студентами навчальної інформації для вирішення учбових задач з метою їх усунення на консультаціях;
- встановлення проміжних рейтингів студентів для самоаналізу, прогнозування майбутнього підсумкового результату та його коректування;
- визначення ефективності розроблених методів і засобів навчання;
- діагностування рівня готовності студентів коледжів та університетів до сприйняття нового матеріалу.

Основними орієнтирами технології оцінювання навчальних досягнень у системі професійної підготовки майбутніх фахівців будівельного профілю стали визначення якості засвоєння навчального матеріалу, ступеня відповідності сформованих умінь та навичок цілям та завданням навчальної дисципліни, встановлення адекватності обраних методів та засобів викладання.

Зазначеним вимогам відповідає рейтингова система оцінювання як складова кредитно-модульної системи організації навчального процесу. Ефективне функціонування рейтингової системи оцінювання знань, як зазначає Л. Романишина, забезпечується урахуванням всіх видів робіт, які виконував студент; оцінюванням ступеня засвоєння дисципліни студентів за загальним рейтингом; можливість перерахування балів рейтингу у звичайну шкалу оцінок; поєднанням системи контролю за навчальною діяльністю із самостійною роботою студентів; орієнтацією на особистість за рахунок індивідуалізації та диференціації навчально-виховного процесу [9, с. 345].

Впровадження педагогічної системи професійної підготовки майбутніх фахівців будівельного профілю передбачало застосування усіх видів контролю (вхідного, поточного, підсумкового та відтермінованого) у коледжах та університетах. З огляду на неоднорідність контингенту студентів (різний рівень підготовки, відмінності у навчальних програмах закладів освіти) під час навчання передбачається адаптаційний період. Тому вважаємо, що у структурі кожної дисципліни повинен бути передбачений вхідний контроль, метою якого є оцінювання базового рівня знань студентів. Така діагностика необхідна як викладачу (для правильної побудови траєкторії викладання дисципліни), так і студенту (для реального самооцінювання власних знань та умінь).

Мета вхідного контролю (тест) – установлення початкового рівня навчальних досягнень студента. Це дає інформацію викладачу про готовність студента до процесу учіння, сприяє диференціації та індивідуалізації навчального процесу.

Поточний контроль здійснювали на кожному занятті на підставі систематичного оцінювання рівня теоретичних знань (тести, усне опитування) та практичних умінь (розв'язання задач, виконання лабораторних робіт). Однією із складових поточного контролю є самоконтроль – оцінювання самим студентом своєї навчальної діяльності. Для проведення самоконтролю потрібно забезпечити студента еталоном навчальної діяльності та очікуваного результату (зразок виконання завдання, ключ до розв'язування задачі можна запропонувати в методичних рекомендаціях, навчальних комп'ютерних програмах).

Підсумковий контроль здійснювали у вигляді модульних контрольних робіт та екзамену у ВНЗ або підсумкових контрольних робіт у коледжах (з дисциплін, де навчальним планом не передбачено екзамену).

Відтермінований контроль (визначення рівня навчальних досягнень через деякий період після підсумкового контролю) проводили письмово за методикою проведення екзамену. За результатами робили висновки про ефективність організації навчально-виховного процесу та запропонованих методик проведення занять.

Використовували усну, письмову та практичну форми перевірки рівня навчальних досягнень студентів. До методів усної перевірки належать індивідуальне опитування, бесіда, до методів письмового контролю рівня навчальних досягнень – предметні (хімічні, фізичні, математичні) диктанти, тести, контрольні роботи, індивідуальні навчально-дослідні роботи. Практична перевірка пов'язана з оцінюванням рівня виконання лабораторних та практичних робіт студентами. Застосування тільки традиційних форм і методів контролю до оцінювання навчальної діяльності студентів виражається в оцінюванні усних та письмових робіт студента, що створює постійну загрозу необ'єктивності. Тотальний перехід на тестові технології діагностики знань призводить до нівелювання творчого підходу, нестандартних рішень, індивідуального темпу і способу розв'язання поставлених завдань. Тому ми переконані у необхідності поєднання усіх способів діагностики знань. Вказані аспекти є ключовими при розробці системи педагогічного контролю рівня навчальних досягнень студентів.

Створення тесту – складна і кропітка робота, яка має певний алгоритм і вимагає високого рівня як предметних, так і педагогічних знань. Завдання для тестового контролю навчання студентів, за Ю. Романенком, мають бути:

- правильними – не допускати вільного тлумачення в тексті завдань;
- однозначними – виключати можливість формулювання багатозначних відповідей;
- відносно короткими – обмежувати кількість слів у тексті завдання;
- технологічними – забезпечувати можливість співвіднести кількісну оцінку виконання тесту із шкалою вимірювання та давати змогу здійснювати математичну обробку результатів;
- придатними – забезпечувати вимірювання рівня навченості широкого контингенту студентів, які оволодівають однаковим обсягом знань на однаковому рівні навчання [8].

Наведені вимоги застосовані нами при розробці різнорівневого тестового контролю на різних етапах ступеневого навчання. При структуруванні системи контролю нами враховано його складові: мету, об'єкт, критерії, кореляцію. Розроблена технологія тестової діагностики рівня знань студентів, що враховує вимоги кредитно-модульної системи навчання, впроваджена у процес викладання природничонаукових дисциплін Луцького національного технічного університету та його структурних підрозділів – Ковельського промислово-економічного та Любешівського технічного коледжів, а також інших ВНЗ – учасників педагогічного експерименту. Контроль відбувається на чотирьох рівнях: вступний контроль, який визначає базовий рівень знань студентів з природничонаукових дисциплін; поточний контроль, що використовується на заняттях для поточної оцінки знань та корекції навчального процесу; підсумковий контроль – модульні контрольні роботи (дві у семестрі), що включають в себе завдання усіх поточних тестів, що проводилися у модулі; завершальний контроль – екзаменаційні завдання та тести перевірки залишкового рівня знань (включають усі тестові завдання дисципліни).

У тестах перших двох рівнів використовуємо завдання множинного вибору або незавершеної думки, що містять базові теоретичні поняття, основні формули та закономірності даної теми. При модульному контролі поряд з тестами, що містять відповіді на питання, використовуємо задачі відкритого типу, розв'язання яких демонструє вміння мислити, проявляти нестандартний підхід до розв'язання типових завдань.

Поряд із безсумнівними перевагами тестовий контроль не позбавлений і низки недоліків, оскільки за цифрами відповіді тесту викладач не бачить індивідуальності студента, його творчих здібностей. Тому переконані у необхідності збереження поряд з тестуванням таких традиційних форм оцінювання знань, як співбесіда, усне опитування, індивідуальні письмові контрольні роботи. Природничонаукові дисципліни (фізика, хімія) передбачають виконання ряду лабораторних робіт. Їх оцінювання проводимо наступним чином: 25 % максимального балу студент отримує за підготовку до лабораторного заняття (записи у лабораторному журналі та розуміння ходу проведення лабораторної роботи, знання властивостей речовин та принципу дії приладів, що використовуватимуться, тощо), 25 % – за проведення лабораторної роботи, дотримання правил техніки безпеки, опис отриманих результатів, 50 % – за захист лабораторної роботи (знання основних теоретичних положень даної теми, аналіз отриманих результатів, вміння розв'язувати задачі за даною темою).

Паралельно із обов'язковими формами поточного контролю знань (виконання тестів, захист лабораторних робіт) практикуємо оцінювання участі студентів у проведенні проблемних лекцій та семінарів, виконанні проєктів, у роботі студентського наукового гуртка, участі в предметних олімпіадах та конкурсах. Таким чином оцінюється кожен вид навчальної діяльності студента незалежно від його базового рівня знань, а пасивність, безініціативність та лінь є причиною низьких семестрових рейтингів окремих студентів.

Формування компетентності нероздільно пов'язане із вмінням розв'язувати різнопланові задачі, до яких висуваються особливі вимоги. У загальнонауковому значенні під задачею розуміють поставлену мету, доручення, завдання, проблему, питання, що вимагають розв'язання тощо. З методичної точки зору, важливим є не тільки результат розв'язку задачі, а сам процес, оскільки саме в процесі розв'язування формується спосіб дії. Професійне спрямування природничонаукових дисциплін реалізується в основному через систему спеціальних міжпредметних задач. Міжпредметними вважають такі задачі, умова, зміст і процес розв'язання яких інтегрує структурні елементи знань про явища природи і суспільства, що вивчаються в різних дисциплінах.

Хімія, фізика, вища математика є природничонауковими дисциплінами у підготовці майбутніх будівельників, їх вивчають на першому курсі як коледжу, так і університету. Враховуючи найтісніший зв'язок із такими спеціальними дисциплінами, як «Будівельне матеріалознавство», «Інженерна геологія і основи механіки ґрунтів», «Технологія будівельного виробництва» та ін., викладачі повинні використовувати на заняттях професійно спрямовані задачі, що містять проблемні елементи. Доцільно впроваджувати задачі, у яких поєднано класичні фізичні та хімічні поняття, закони і речовини, що застосовуються у будівництві, з використанням необхідного для розв'язування математичного апарату. Для прикладу наводимо ряд таких задач. Оскільки для розв'язання наведених задач потрібні мінімальні знання та навички з фізики та хімії та базові вміння з математики, то їх можна використовувати як на уроках у технічному училищі чи коледжі, так і на заняттях в університеті.

Муфельна піч, що використовується для отримання вапна, споживає потужність 1 кВт. Температура її внутрішньої поверхні при відкритому отворі площею 25 см^2 дорівнює $1,2 \text{ КК}$. Вважаючи, що отвір печі випромінює, як чорне тіло, визначити, яка частина потужності розсіюється стінками.

При вертикальному підйомі вантажу масою 200 кг підйомним краном на висоту 10 м постійною силою F була виконана робота $A=784 \text{ кДж}$. З яким прискоренням піднімали вантаж?

Кальцит, або вапняковий шпат, CaCO_3 – один з найпоширеніших мінералів земної кори. Для розпізнавання карбонатних гірських порід використовують розчин хлоридної кислоти. Записати рівняння реакції, яка проходить при їх взаємодії. Визначити об'єм вуглекислого газу, що виділиться при розчиненні у розчині хлоридної кислоти 100 г кальциту, який містить 5% нерозчинних домішок.

Натрій оксид прискорює процес варіння скла. Яку масу соди потрібно взяти для утворення 100 кг скла із вмістом натрій оксиду 10% ?

Реакція дегідратації гіпсу $\text{CaSO}_4 \cdot 2\text{H}_2\text{O}$ відбувається з поглинанням тепла. Для отримання 1 кг напівводного гіпсу $\text{CaSO}_4 \cdot 0,5\text{H}_2\text{O}$ теоретично потрібно затратити 580 кДж тепла. Визначити тепловий ефект цієї реакції, записати термохімічне рівняння.

Диференціальні рівняння дають можливість розв'язувати багато задач технології будівельного виробництва, зокрема, міцності опор мостів. Знайти загальний розв'язок рівняння:

$$2x \, dx + 2y \, dy = 0.$$

Упровадження в навчальний процес професійно спрямованих задач має низку позитивних моментів. Такі задачі – потужний засіб виховання інтелектуальної культури та грамотності студентів (формування ключових компетентностей), оскільки вони ілюструють необхідність отримуваних знань та умінь у практиці роботи інженера-будівельника. Це спонукає до активного вивчення природничонаукових дисциплін, розвиває навички свідомого і раціонального використання хімічних, фізичних, математичних знань та умінь у вивченні дисциплін професійного та практичного спрямування, а пізніше і у продуктивній діяльності за спеціальністю будівельного профілю.

На основі визначених рівнів сформованості професійної компетентності майбутніх будівельників розроблено чотирирівневі задачі з природничонаукових дисциплін, послідовне розв'язування яких протягом навчального модуля дозволяє ступенево формувати когнітивний та діяльнісний компоненти компетентності. Наводимо приклад задачі з хімії.

Умова (теоретичні відомості): При виробництві клінкеру використовують два основні природні матеріали – вапняк та глину. Температура відпалу досягає 900–1000 °С. Утворюється побічний продукт – вуглекислий газ.

Рівень I (репродуктивний): записати хімічні формули, що відображають основний склад вказаних природних матеріалів; зазначити клас неорганічних сполук, до якого вони належать; використовуючи довідник з хімії, описати фізичні властивості цих сполук (матеріалів).

Рівень II (реконструктивний): записати рівняння хімічних процесів, що відбуваються при виробництві клінкеру; вказати типи хімічних реакцій.

Рівень III (продуктивний): розрахувати теплові ефекти реакцій, що відбуваються; обґрунтувати необхідність проведення технологічного процесу при високій температурі.

Рівень IV (творчий): оцінити переваги та недоліки процесу виробництва клінкеру; вказати фактори інтенсифікації виробництва, виокремити ризики для навколишнього середовища.

Таке структурування професійно спрямованих задач з природничо-наукових дисциплін дозволяє застосовувати їх у навчальному процесі не тільки ВНЗ, а й коледжів (студенти виконують завдання I-II рівнів та творче завдання відповідно до обсягу здобутих знань). Випусники коледжів (ПТНЗ), що продовжують навчання у ВНЗ, нерідко демонструють глибше розуміння специфіки будівельного виробництва, можуть краще визначити переваги чи недоліки окремих технологічних процесів, оскільки вже знайомі з ними на практиці. Завдання четвертого рівня оцінюються найвище, тому це дозволяє діагностувати не тільки когнітивний компонент, але й творчий підхід.

Проблема трансформації здійснення екзаменаційного контролю особливо загострилася при впровадженні кредитно-модульної організації навчального процесу. З'явилися два діаметральні погляди: перший – екзаменаційний контроль трансформується у підведення рейтингових балів, а іспит можна не складати; другий – екзамен повинні складати всі студенти, бо це єдина форма адекватного оцінювання рівня навчальних досягнень. На наше переконання, істина знаходиться посередині.

За провідним компонентом (теорія І. Лернера) навчальні предмети поділяють на три типи: навчальні предмети з провідним компонентом «наукові знання», або основи наук; навчальні предмети з провідним компонентом «способи діяльності»; навчальні предмети з провідним компонентом «художня освіта і морально-естетичне виховання», або «естетичне бачення світу». Ця типізація стала основою для обґрунтування вибору способів екзаменаційного контролю для навчальних предметів першого типу, до яких належить фізика, хімія, вища математика, Розглянемо детальніше методику проведення екзаменаційного контролю з природничонаукових дисциплін у вищих навчальних закладах. У минулі роки на екзамені використовували ряд екзаменаційних білетів (25–30 варіантів), що містили, як правило два теоретичні питання та розрахункову задачу або вправу. Екзамен студенти складали усно, попередньо підготувавшись до відповіді протягом 30–45 хв. Викладач заслуховував усіх студентів по черговому, що приводило до того, що екзамен тривав нерідко 6–8 годин.

При впровадженні кредитно-модульної системи екзамен став обов'язковою формою контролю, однак, за досвідом викладацької діяльності, лише приблизно третина студентів не складають екзамену. На нашу думку, спосіб проведення екзамену повинен корелювати із модульними контрольними роботами, оскільки на основі їх написання, а також поточного рейтингу студент має можливість отримати семестрову оцінку без складання іспиту.

Наведемо методику проведення екзамену з природничонаукових дисциплін, яку практикуємо під час навчання майбутніх будівельників. Екзамен проходить у два етапи. На першому усі студенти групи одночасно письмово виконують завдання екзаменаційного білету (які можна проводити і у формі комп'ютерного контролю, що суттєво економить час викладача на перевірку робіт та виключає суб'єктивний фактор). На другому етапі викладач проводить співбесіду за теоретичним питанням білета або, за потреби, за додатковими питаннями. Якщо студента влаштовує набрана сума балів за письмове виконання – співбесіда не проводиться.

Екзаменаційний білет містить 3 групи завдань, кожна з яких оцінюється у 20 балів: I – тестові завдання, що охоплюють базові поняття теоретичного матеріалу курсу (навести правильну відповідь, не наводячи розв'язок); II – розрахункові задачі професійного спрямування (навести повний розв'язок та відповідь); III – теоретичне питання, на яке студент дає усну відповідь. Застосування екзаменаційних білетів розробленої структури дозволяє максимально об'єктивно оцінити навчальні досягнення студента. Це досягається поєднанням об'єктивності тестових

технологій, застосуванням діяльнісних методик розв'язання задач та міжособистісною взаємодією під час усної співбесіди.

Авторський підхід до складання іспитів відрізняється від традиційного, за якого оцінювання знань мало суб'єктивний характер, нерідко містило елементи оцінювання особистості студента та його поведінки, а не навчальних досягнень. Чітке обґрунтування отриманих балів за виконанні завдання на екзамені знижує емоційну напругу як студентів, так і викладача, сприяє уникненню конфліктних моментів і об'єктивній оцінці навчальних досягнень.

Система педагогічного контролю впроваджена в навчальний процес у Луцькому національному технічному університеті та його структурних підрозділах – Ковельському промислово-економічному та Любешівському технічному коледжах. Результати зимової сесії з дисциплін природничонаукової підготовки (фізика, хімія) 2015/16 н.р. студентів напрямку «Будівництво» Луцького національного технічного університету показали, що 32,8 % студентів отримали позитивні оцінки за результатами рейтингу, не складаючи екзамену, 35,7 % – на екзамені покращили свій рейтинг, 9,9 % студентів отримали позитивну екзаменаційну оцінку після повторного складання іспиту, 2,6 % – мали незадовільні успіхи (менше 35 балів), були відраховані або спрямовані на повторне вивчення дисципліни. Деталізовані результати іспиту за основною відомістю наведено у табл. 1. Очевидно, що європейська шкала ECTS, яка має сім рівнів оцінювання, краще відображає навчальні досягнення студентів.

Таблиця 1

Результати складання іспиту з хімії та фізики студентами напрямку «Будівництво»

Оцінка за державною (національною) шкалою	відмінно	добре		задовільно		незадовільно	
Оцінка за шкалою ECTS	A	B	C	D	E	FX	F
Кількість балів	90–100	84–89	75–83	64–74	60–63	35–59	0–34
Відносна кількість студентів, %	18,5	3,6	31,3	10,0	24,1	9,9	2,6

Порівняльний аналіз вхідного та підсумкового контролю навчальних досягнень студентів з природничонаукових дисциплін (фізика, хімія) наведено на рис. 1. Погоджуємося із думками викладачів інших технічних ВНЗ України щодо низького рівня базової підготовки студентів у середніх навчальних закладах. Однак із запровадженням незалежної системи оцінювання знань випускників спостерігаємо тенденцію до покращення рівня знань студентів внаслідок жорсткішої системи відбору абітурієнтів при вступі до вищих навчальних закладів.

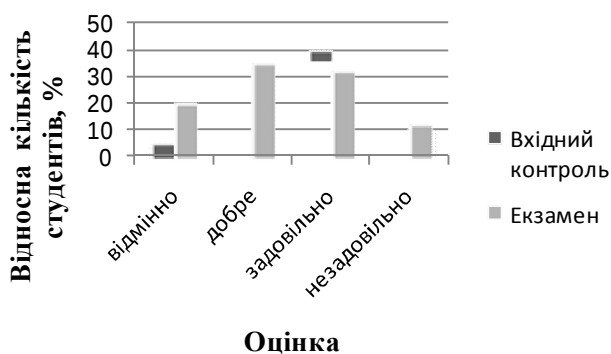


Рис. 1. Порівняльні результати вхідного та підсумкового контролю знань з природничонаукових дисциплін студентів напрямку «Будівництво».

Студенти, що навчаються на державній формі навчання, які мали високі рейтингові бали незалежного оцінювання, демонструють суттєво кращі результати навчальної діяльності, ніж їх одногрупники контрактної форми. Під час проведення вхідного контролю знань студенти державної форми навчання показали в основному відмінні та добрі результати, тоді як студенти контрактної форми – задовільні, а то і незадовільні. Така ж тенденція збереглася і при підсумковому контролі.

Висновки та перспективи подальших досліджень. Компетентнісний підхід визначає результативно-цільову спрямованість освіти, що, на нашу думку, є його безперечною перевагою над іншими традиційними та інноваційними підходами. Процес моніторингу повинен забезпечити постійне відстежування, аналіз та звітування про компетенції, які поступово формуються у студента протягом навчання. Оцінка рівня компетентностей – складна педагогічна проблема, одним із шляхів розв'язання якої є запропонована технологія педагогічного контролю навчальних досягнень студентів. На основі результатів її впровадження у Луцькому НТУ, його структурних підрозділах та залучених до експерименту ВНЗ можна зробити наступні висновки:

- кредитно-модульна технологія з рейтинговою системою оцінювання сприяє підвищенню якості навчання за умови сумлінної роботи викладача і належного рівня знань студента;
- систематичний контроль і консультації сприяють поліпшенню якості навчального процесу, в результаті якого навіть студенти із низьким базовим рівнем знань отримують задовільну оцінку;
- екзамен як форма контролю навчальних досягнень студентів не втратив своєї актуальності, а набув нового змісту;
- адекватне оцінювання знань студентів досягається лише використанням різнопланових методів оцінки (тестів, усних співбесід, розрахункових робіт);
- дана технологія навчання вимагає високої кваліфікації і досвіду викладачів та значно більших затрат часу на індивідуальну роботу зі студентами;
- діагностика якості знань за цією системою здійснюється з використанням комплексу навчально-методичних матеріалів.

Подальші дослідження спрямовуватимемо на залучення студентів до створення системи контролю навчальних досягнень.

1. Аванесов В. С. Определение исходных понятий теории педагогических измерений / В. С. Аванесов // Педагогические измерения. – 2005. – № 2. – С. 17–20.
2. Беспалько В. П. Инструменты диагностики качества знаний учащихся / В. П. Беспалько // Школьные технологии. – 2006. – № 2. – С. 138–150.
3. Горбатюк Р. Експериментальна модель дистанційного навчання майбутніх фахівців у вищому навчальному закладі / Р. Горбатюк, Л. Романишина // Наукові записки Тернопільського національного педагогічного університету імені Володимира Гнатюка. Серія: педагогіка. – 2016. – №. 2. – С. 68–75.
4. Дорошенко Ю. О. Достоірність комп'ютерного тестування : навч.-метод. посібн. / Ю. О. Дорошенко, П. А. Ротаєнко. – К. : Педагогічна думка, 2007. – 176 с.
5. Лукіна Т. О. Технології діагностики та оцінювання навчальних досягнень: навч.-метод. матеріали / Т. О. Лукіна. – К., 2007. – 62 с.
6. Ляшенко О. І. Тестові технології в системі освіти України: стан і перспективи розвитку [Електронний ресурс] / О. І. Ляшенко, С. А. Раков. – Режим доступу: monitoring.in.ua/up/files/publikacii/Ukraine/testovitehnologii_v_osviti.pdf.
7. Рач В.А. Контроль і моніторинг у реалізації освітніх проєктів/ В.А. Рач, А.Ю. Борзенко // Управління проєктами та розвиток виробництва: зб. наук. пр. – Луганськ: вид-во СНУ ім. В. Даля, 2005. – № 2 (14). – С. 72–81.
8. Романенко Ю. А. Інструментарій моніторингу: принципи та вимоги / Ю. А. Романенко // Нові технології навчання: наук.-метод. зб. – К.: Інститут інноваційних технологій і змісту освіти МОН України, 2008. – Вип. 53. – С. 3–8.
9. Романишина Л. М. Система поетапного контролю навчальної діяльності студентів педагогічних університетів за модульно-рейтинговою технологією навчання з дисциплін природничого циклу : дис... д-ра пед. наук : 13.00.04 / Романишина Людмила Михайлівна. – К., 1997. – 417 с.
10. Смолінчук Л. С. Тестування як метод оцінювання навчальних досягнень студентів // Вісник Національного авіаційного університету. Серія: Педагогіка, Психологія. – 2012. – Т. 1. – № 3. – С. 74–79.
11. Тверезовська Н. Т. Інноваційний підхід до іспиту як форми контролю знань студентів / Н. Т. Тверезовська // Педагогіка : зб. наук. праць Бердянського держ. пед. ун-ту (Педагогічні науки). – 2008. – № 4. – С. 205–210.
12. Челышкова М. Б. Теория и практика конструирования педагогических тестов / М. Б. Челышкова. – М. : Логос, 2002. – 432 с.
13. Шахов В. Концепція базової педагогічної освіти майбутніх учителів / Володимир Шахов // Особистісно-професійний розвиток майбутнього вчителя : Монографія [Шестопалюк О.В. (кер), Акімова О.В., Галузяк В.М. та ін.]. – Вінниця: Тов. «Нілан ЛТД», 2014. – С. 346–374.
14. Hernández R. Students' Engagement in the Development of Criteria to Assess Written Tasks. – From the REAP International Online Conference on Assessment Design for Learner Responsibility, 2007. – Retrieved from <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

УДК 371.315

Дудка О.М. к.п.н., Власій О.О., к.т.н.

ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»

ОСОБЛИВОСТІ ВИВЧЕННЯ ПРОГРАМУВАННЯ НА SCRATCH

Дудка О.М., Власій О.О. Особливості вивчення програмування на Scratch. В статті розглянуто особливості процесу вивчення програмування як навчальної дисципліни; проведено аналіз можливостей візуальної мови програмування Scratch з точки зору сучасних парадигм програмування; запропоновано розробку ігрових Scratch-проектів як мотиваційний засіб при вивченні програмування; проаналізовано наявні дидактичні матеріали для вивчення Scratch; запропоновано використання навчального сайту для зацікавленості програмуванням в Scratch.

Ключові слова: мова програмування, Scratch, комп'ютерна гра, Scratch-проект, навчальний сайт

Дудка О.М., Власій О.О. Особенности изучения программирования на Scratch. В статье рассмотрены особенности процесса изучения программирования как учебной дисциплины; приведен анализ возможностей визуального языка программирования Scratch с точки зрения современных парадигм программирования; предложена разработка игровых Scratch-проектов как мотивационное средство при изучении программирования; проанализированы имеющиеся дидактические материалы для изучения Scratch; предложено использование учебного сайта для заинтересованности программированием в Scratch.

Ключові слова: язык программирования, Scratch, компьютерная игра, Scratch-проект, навчальний сайт

O. Dudka, O. Vlasii Improving introductory programming with Scratch. The characteristic of learning programming in secondary school is given. The capabilities of visual programming language Scratch in terms of modern programming paradigms are analyzed. The development of game Scratch-projects as a motivational tool in the study of programming is proposed. Existing teaching resources for learning Scratch are analyzed. Training site about Scratch as interest-based portals is proposed.

Ключові слова: programming language, Scratch, computer game, teaching site

Постановка проблеми. Вивчення програмування в наш час стало одним із пріоритетних напрямків підготовки майбутніх професіоналів в ІТ-галузі. Цікаво зауважити, що з'явилася значна кількість інтернет-ресурсів, які пропонують вивчення різних мов програмування за різною методикою для різних вікових категорій. Деякі ресурси пропонують вивчати ази програмування, а вірніше ази алгоритмізації, користувачам «які ще не вміють читати» (<https://code.org/>).

Процес вивчення програмування у середній школі є актуальною проблемою сьогодення. Внаслідок великого розмаїття навчальних програм, технічних можливостей шкіл, рівнів підготовки педагогів на даний момент не існує єдиного підходу до узгодженого централізованого вивчення конкретної мови програмування. Думки методистів, ІТ-експертів та програмістів щодо вибору мови програмування, яку необхідно вивчати у середній школі, кардинально розходяться. Одні відстоюють класичну версію навчальної мови Pascal, інші – візуальну реалізацію Lazarus, ще інші – сучасні C++ та Python, деś між пропозицій, звісно, є і HTML [1, 2]. Хочеться зауважити, що про підготовку до класичного програмування, так би мовити, початковий рівень програмування, таких дискусій не ведеться. Тут першість займає, безумовно, Scratch – візуальна мова програмування. Scratch можна розглядати в певній мірі як прототип Flash, хоча Scratch появився значно пізніше. Можливо, саме здатність поєднання графіки, медіа та програмування зі зручним користувацьким інтерфейсом і підштовхнуло до створення чогось аналогічного, але простішого і доступнішого, саме для дітей. Скептики, які вважали Scratch іграшкою, недостойною уваги, все ж змушені були визнати, що за допомогою саме цієї мови і цього середовища дітям набагато простіше даються базові уявлення про програмування. Йде мова саме про розвиток алгоритмічного мислення, яке є необхідним фундаментом будь-якого процесу розробки програмного забезпечення. Scratch дав у руки потужний інструмент створення мультимедійних інтерактивних проектів на основі реалізації складених алгоритмів у вигляді скриптів (так називають програми на мові Scratch). Тому метою дослідження є аналіз можливостей використання Scratch як початкового етапу у вивченні програмування та створення ефективної інформативної системи для надання можливості учням додатково вивчати Scratch, а завданням – популяризація програмування серед учнів, поєднання навчання з ігровими ситуаціями, мотивація школярів до розробки ігрових проектів.

Аналіз можливостей Scratch. Scratch створено під керівництвом професора Митчела Резника (Mitchel Resnick) у дослідницькій групі Lifelong Kindergarten research group при Массачусетському технологічному інституті. Попередня відома розробка цієї групи — це

популярна серія LEGO Mindstorms, що є комплектом деталей для складання програмованих роботів. Робота велася у співдружності з дослідниками Університету Лос-Анджелеса (UCLA — University of California, Los Angeles) при фінансовій підтримці Національного наукового фонду і Фонду корпорації Intel. Scratch – вільне програмне забезпечення (ліцензія GNU GPL), до переваг якого варто виділити кросплатформенність, наочність запису алгоритму, багатомовність інтерфейсу (кілька десятків мов) [3, 4]. Scratch знайомить з базовими поняттями алгоритмізації та програмування: типи даних (символьні, числові, логічні, графічні, аудіо), змінні (глобальні й локальні), операції (арифметичні, логічні), вирази (числові, текстові, логічні, порівняння), оператори, функції, алгоритмічні структури (лінійні, розгалужені, циклічні), введення й виведення даних. Scratch можна вільно використовувати у шкільній чи позашкільній освіті, скачавши середовище програмування з офіційного сайту розробників <http://scratch.mit.edu/>.

Ігрові проекти як мотиваційний засіб вивчення програмування. Комп'ютерні ігри – з недавніх пір це словосполучення міцно увійшло в наше життя: кожен, хто має комп'ютер, напевно, зміг відчути їх привабливість, мабуть, гра закладена в саму природу людини. Комп'ютерні ігри відносяться до числа актуальних проблем сучасної науки. Ігри допомагають нам змодельовати різні життєві ситуації, проблеми та надають деякі можливі шляхи їх вирішення. Гра містить в собі всі необхідні передумови для природного розвитку особистості і культури суспільства. З вдосконаленням комп'ютерів удосконалювалися й ігри, залучаючи все більше і більше людей. На сьогоднішній день комп'ютерна техніка досягла такого рівня розвитку, що дозволяє програмістам розробляти дуже реалістичні ігри з хорошим графічним і звуковим оформленням. З 2011 року комп'ютерні ігри офіційно визнані в США окремим видом мистецтва. Комп'ютерні ігри зробили настільки істотний вплив на суспільство, що в інформаційних технологіях відзначена стійка тенденція до «гейміфікації» для неігрового прикладного програмного забезпечення. Думка, що ігри роблять лише для дітей, не є вірною. 75% цільової аудиторії ігрового бізнесу – це люди, старші за 35 років. Розробка ігор – це точка перетину мистецтва та високих технологій. В компаніях, що працюють в ігровому бізнесі, лише 20-25% роботи виконують програмісти. Все інше – це праця людей нетехнічної спеціалізації, таких як контент-менеджери, маркетингологи, художники чи дизайнери, і їхній внесок стає все вагомим [5, 6].

Багато розробників розглядають індустрію комп'ютерних ігор лише як сектор економіки, в якому можна непогано заробити. Але, насправді, створення ігор – це ще й інструмент культури. Не секрет, що для сучасних дітей часто ігри заміняють дитячі казки, які формують світогляд, для підлітків вони створюють персонажів-кумирів, замінюючи собою класичні книги, і лише для дорослих людей ігри служать здебільшого тільки розвагою. При правильному підході ідеї, закладені в комп'ютерні ігри, дуже добре засвоюються дітьми. Сучасні діти пізнають світ багато в чому через призму комп'ютерних ігор. Ігри ці створюються людьми, часто далекими від культури, педагогіки, психології. А в деяких випадках ігри навмисно створюються з метою пропаганди тих чи інших далеко не позитивних ідей. Культурний потенціал комп'ютерних ігор ще тільки починає розкриватися, але вже зараз зрозуміло, що потенціал цей – величезний [7].

Тому аж ніяк не можна заперечувати значущість комп'ютерних ігор, а навпаки – зважаючи на їх прихований потенціал, використовувати комп'ютерні ігри з метою виховання, навчання та розвитку. На нашу думку, можливості Scratch кардинально змінюють мотивацію вивчення програмування. Якщо при вивченні алгоритмів (наприклад, Logo) в основі мотивації лежить потреба складання алгоритму для виконання головним героєм певного набору дій (хай це Черепашка, Кенгуру чи Садівник), то за можливості реалізації в Scratch мультимедійного проекту за власним сценарієм (від розробки головного героя до програмування його довільних дій) на чільне місце виходить бажання реалізувати власні фантазії, що, безумовно, є потужним інструментом мотивації. Цікаво зауважити, що при цьому завдання формулюються як завдання гри, а не як завдання з вивчення алгоритмів. Потреба володіння знаннями про ті чи інші команди та алгоритми виникає з природного бажання реалізувати задумане і є невидимим потужним інструментом мотивації.

Розробка гри на прикладі Scratch-проекту. Візуальна мова програмування Scratch, хоча розглядається як мова програмування для дітей та підлітків, має широкі можливості для реалізації прототипів повноцінних ігрових програм [8]. Scratch дає можливість легко поєднувати графіку, зображення, звук, а саме такі елементи поєднує в собі будь-яка комп'ютерна гра. Процес розробки ігрового проекту в середовищі Scratch включає найважливіші характерні етапи створення

комп'ютерної гри: від ідеї та пошуку жанру до публікації та документування. Наведемо лише кілька аналогій: вмонтований графічний редактор дає можливості розробляти графічні образи героїв та створювати дизайн сцен; в свою чергу, розробка образів спрайтів ознайомлює із основами класичної анімації; послідовне виконання скриптів, організоване на основі передачі повідомлень, дає поняття про механізми передачі параметрів; паралельне виконання скриптів дає уявлення про паралельні процеси; можливість використання датчиків підводить до розуміння інтерактивності; можливість прив'язки змінних та скриптів до спрайтів – до поняття об'єктно-орієнтованого програмування. Як вже зауважувалося вище, Scratch є відмінним інструментом для створення комп'ютерних ігор.

Розглянемо етапи розробки типової комп'ютерної гри:

1. Ідея.
2. Сценарій: герої, їх ролі, правила, за якими вони взаємодіють, і головне - мета гри!
3. Створення чи пошук необхідних зображень та звуків.
4. Написання скриптів для спрайтів та сцен: реалізація самостійних дій спрайтів (паралельні скрипти), реалізація взаємодії спрайтів (послідовні скрипти), якщо гра багатоскладова чи багаторівнева – забезпечення переходу між її складовими чи рівнями.
5. Написання правил та довідок для користувача.
6. Тестування проекту.
7. Виправлення помилок.
8. Створення опису гри.
9. Публікація проекту.

Розглянемо ілюстративний приклад процесу розробки комп'ютерної гри на всіх його етапах:

- Ідея: є предмети двох типів, необхідно посортувати, який до якого типу відноситься.
- Можливий сценарій: Рудий Кіт пропонує гравцеві посортувати розкидані по екрану об'єкти.
- Варіанти спрайтів: головний герой (наприклад, Рудий кіт).
- Об'єкти двох типів (наприклад, квадрати і кружки, овочі і фрукти, голосні і приголосні букви, птахи і тварини - кому що до вподоби).
- Контейнери для сортування (наприклад, ящики, кошики, хатинки, книжки і т.п.).
- Вибір ролей (наприклад Рудий кіт – пропонує гравцеві посортувати об'єкти, може допомагати підказками. Об'єкти – хаотично розкидані по екрану; можуть перетягуватися мишкою по екрану; зникають, коли доторкаються до потрібного контейнера. Контейнери – мають визначене місце; видають різні звуки в залежності, чи до них доторкається правильний чи неправильний об'єкт).

Звісно, ролі можна вдосконалювати, ускладнювати сценарій додатковими можливостями, але порада: спочатку краще спробувати реалізувати мінімальний "кістяк" проекту, а потім розробляти додаткові можливості. Діємо за правилом: не все одразу!

Зауважимо, що для зображення взаємозв'язку та взаємодії об'єктів можна використовувати діаграми, які дозволяють наочно зобразити властивості та взаємозв'язки об'єктів і які можна вважати сходинкою до освоєння в майбутньому діаграм потоків та подій (див. рис. 1).

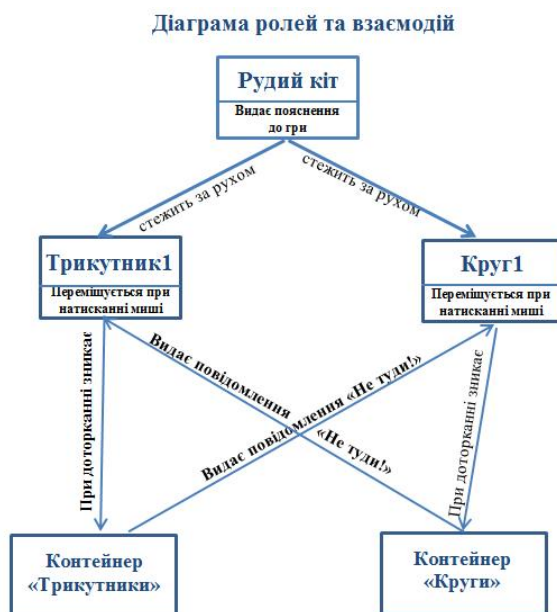


Рис.1 Діаграма ролей та взаємодій об'єктів у ігровому Scratch-проекті

Навчальний сайт як портал зацікавленості Scratch. Навчальні програми XX pp.. передбачали вивчення Scratch у 5-7 класах, діючі навчальні програми переносять вивчення Scratch у початкову школу. Однак практика показує, що бажання дітей реалізувати свої ідеї на Scratch значно випереджають програму, а найчастіше виходять за її рамки. В першу чергу йде мова про реалізацію комп'ютерних ігор. Не секрет, що залежність від комп'ютерних ігор стала однією із серйозних проблем сучасності. То чому ж таку залежність не використати в позитивному плані? Коли дитина бачить, що вона не просто може грати, а може сама розробити гру і дати завдання своїм друзям, знайомим та навіть і вчителям, то з геймера вона перетворюється на розробника програмного забезпечення, який має практичний досвід роботи з ігровими програмами і при цьому хоче втілити особисті ідеї на практиці.

Повноцінним інструментом для розробки комп'ютерної гри, звичайно, є Scratch. Тому актуальною проблемою є створення ефективної інформативної системи для надання можливості учням додатково вивчати візуальну мову програмування Scratch. Друкованих посібників зі Scratch практично немає (окрім, звісно, підручників, в яких вивчення Scratch по краплинах вивчається в школі, що не дає змоги реалізуватися ідеям творчих учнів). Постійно появляються нові електронні посібники з програмування на Scratch, однак зауважимо, першість тут ведуть англомовні джерела. На сьогодні є багато сайтів для вивчення програмування на Scratch, проте в них є суттєвий недолік – мова інтерфейсу та недостатній рівень методичного викладу матеріалу. На нашу думку, для вивчення Scratch найбільш відповідні електронні джерела, оскільки вивчення Scratch безпосередньо пов'язано із графікою, звуком та анімацією, які набагато цікавіше демонструвати «вживу», аніж на папері.

Для досягнення поставленої мети авторами розроблено сайт «Вивчаємо Scratch разом» [9], який створено за допомогою оновленої версії Google-сайтів (див. рис. 2). На сайті, окрім навчального матеріалу, висвітлюються події, пов'язані із вивченням програмування на Scratch; розміщуються зразки кращих проектів, розроблених учнями; даються цікаві завдання для виконання всім бажаючим; розміщуються приклади реалізації ігрових проектів та створення мультфільмів. На даний момент процес розробки сайту продовжується, зокрема розробляється окрема сторінка сайту, присвячена проектуванню та створенню комп'ютерних ігор. Даний ресурс можна використовувати як для навчання програмуванню Scratch на уроках інформатики, так і для самопідготовки учнів, адже середовище програмування Scratch можна безкоштовно завантажити і вільно використовувати. За допомогою сайту учні мають можливість керувати своїм навчанням, користуватись готовими розробленими сценаріями та опубліковувати власні проекти в міжнародній спільноті в мережі Інтернет (<http://scratch.mit.edu>).

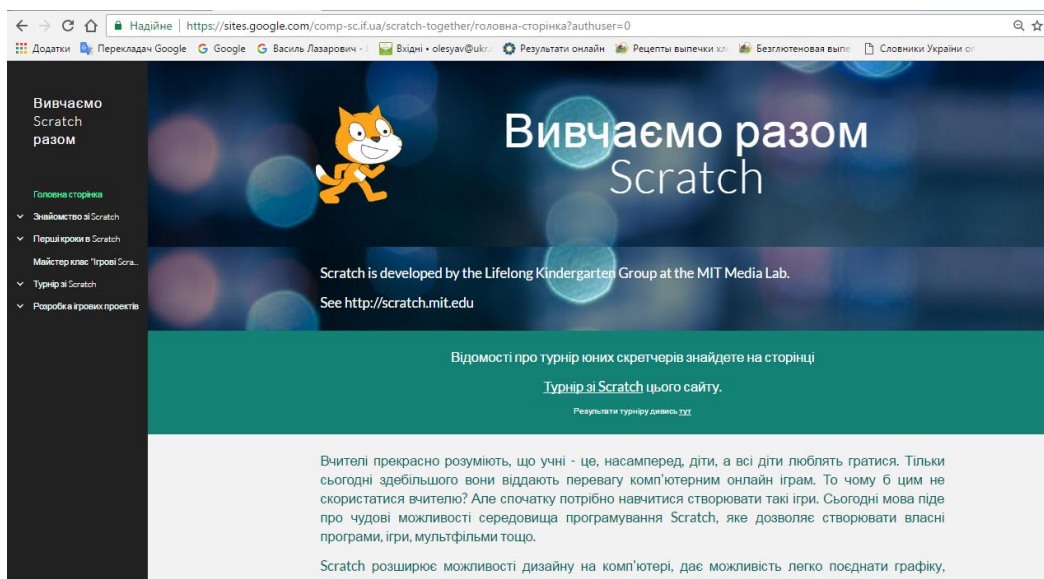
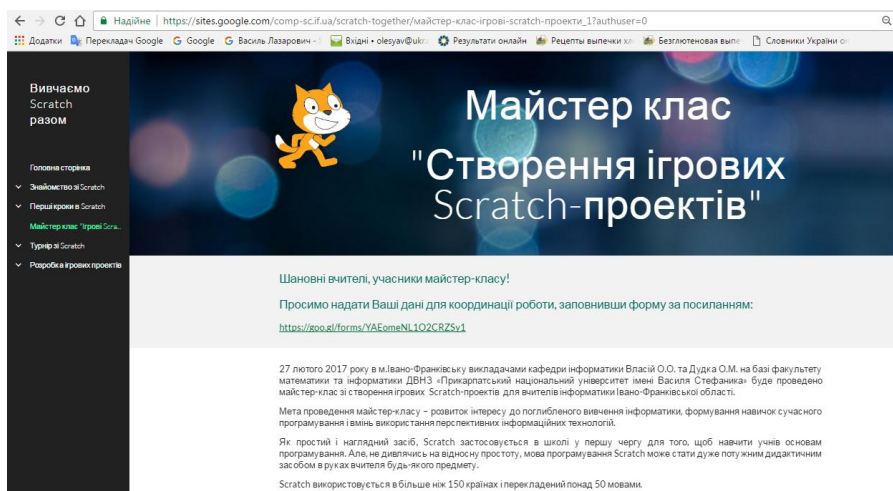


Рис. 2 Навчальний сайт для вивчення Scratch

Сайт має великий потенціал для активного та інтерактивного навчання, розвитку навичок мислення вищого порядку і більшої гнучкості навчального процесу, дозволяє учням використовувати різні способи особистісного та інтелектуального освоєння нових понять в процесі візуалізації абстрактних ідей та інтерактивної взаємодії. Педагогічний потенціал сайтів в першу чергу полягає у тому, що учні в процесі одержання, трансформації знань і подальшому публікуванні своїх робіт вчаться конструювати знання, засновані на відносинах і спілкуванні. Для викладачів опубліковані праці учнів – це можливість зробити висновки про те, як учні трансформують і засвоюють зміст і стратегію. Для бажаючих вчитися подібна публікація – це матеріал для подальшої рефлексії і аналізу, який дозволяє їм ще раз звернутися до своїх робіт і переосмислити їх, збагатити таким чином свій навчальний досвід.

Апробація. В якості апробації отриманих результатів на факультеті математики та інформатики Прикарпатського національного університету імені Василя Стефаника було проведено обласний майстер-клас зі створення ігрових Scratch-проектів, матеріали якого були розміщені на цьому сайті (див. рис. 3). Варто зазначити, що кількість учасників перевищила очікування – близько 70 дітей, причому з різними рівнями знання Scratch, виявили бажання дізнатися про Scratch та понад 20 вчителів інформатики підтримали своїх вихованців у їх прагненнях до вдосконалення знань зі Scratch.



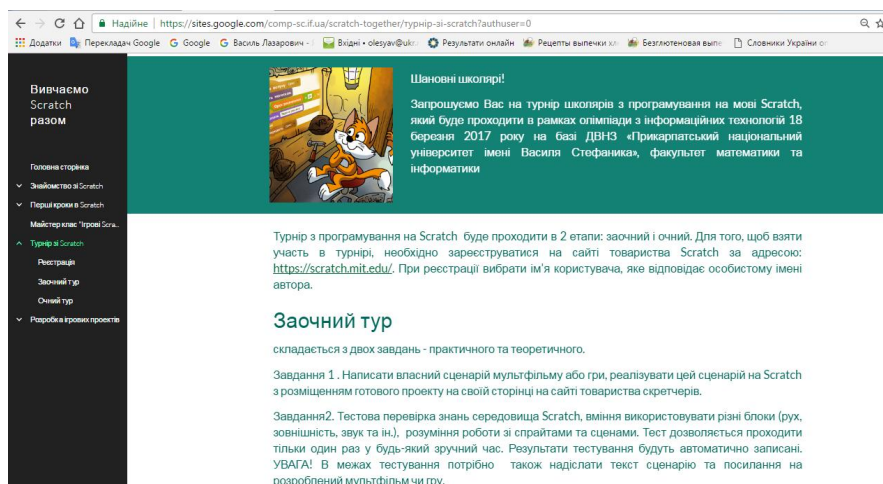


Рис. 3 Використання сайту для організації творчих заходів

З метою раннього залучення молодших школярів до програмування також вперше було проведено обласний турнір юних програмістів на мові Scratch серед учнів 8-12 років, який проходив в 2 етапи: заочний і очний, матеріали якого також розміщені на сайті (див. рис. 3). В заочному турі прийняли участь понад 60 школярів, до очного туру було допущено 40 учасників. Учні проявили значну зацікавленість розробкою ігрових Scratch-проєктів, під час виконання завдань проявляли активність та креативність при реалізації запропонованих ідей.

На даний момент вчителі інформатики Івано-Франківської області використовують розроблений сайт на своїх уроках для навчання програмуванню Scratch, а їх учні – для вдосконалення своїх знань зі Scratch.

Для того, щоб ефективно використовувати сайти у навчальному процесі слід дотримуватися таких рекомендацій: по-перше, викладач повинен ознайомитись з сайтами створеними іншими викладачами з метою формування уявлення про те, як вони можуть використовуватися у навчальному процесі; по-друге, викладач повинен створювати власні сайти, щоб мати уявлення про їх функції і можливості на практиці; по-третє, перед тим як запропонувати такий засіб навчання учням необхідно змоделювати сайт для своїх учнів (розробка правил, обговорення тематики, попередня підготовка матеріалів); по-четверте, створені сайти необхідно популяризувати, щоб відбувався процес обговорення та спілкування.

Розробка педагогами сайтів до певної дисципліни або теми дозволяє здійснювати науково-методичний супровід процесу навчання оперативно, через вчасне інформування та консультування. Сайт розкриває можливості тісного спілкування учителя з учнями. За допомогою різноманітних віртуальних ресурсів учень отримує методичну допомогу в зручний для нього час.

Все це дозволяє вивести сучасне заняття на якісно новий рівень: підвищується статус викладача; розширюються можливості ілюстративного супроводу заняття; використовуються різні форми навчання та види діяльності в межах одного заняття; ефективно організовується контроль знань, вмінь та навичок учнів; полегшується та вдосконалюється розробка творчих проєктів.

Висновки. Візуальна мова програмування Scratch має всі можливості для розробки ігрового програмного забезпечення. Простий і доступний інструментарій роботи в середовищі Scratch робить процес створення ігрового проєкту цікавим та зрозумілим і водночас дає можливість в повній мірі відчувати особливості технологій розробки комп'ютерних ігор. В наші дні програмування нерідко називають другою грамотністю. Його вивчають і в школі, і у вузах, хоча, на жаль, далеко не завжди успішно. Важливість освоєння цієї дисципліни не викликає сумніву з урахуванням тенденцій впровадження інформаційних технологій у різні аспекти нашого життя.

1. Яку мову програмування вивчати у школі (матеріали для дискусії) // Комп'ютер у школі та сім'ї. – 2013. – № 7. – С.14-18. – Режим доступу: http://nbuv.gov.ua/UJRN/komp_2013_7_5.
2. Яку мову програмування вивчати у школі (матеріали для дискусії) // Комп'ютер у школі та сім'ї. – 2013. – № 8. – С. 9-18. – Режим доступу: http://nbuv.gov.ua/UJRN/komp_2013_8_3

3. Шевченко І. С. Використання навчального середовища SCRATCH при вивченні теми "Алгоритми та їх виконавці" / Шевченко І. С. // FOSS Lviv 2014, 24-27 квітня 2014 року — Л., 2014 — С. 114-117.
4. Величко В.Є. Використання технології візуального програмування в університетській освіті засобами вільного програмного забезпечення/ В. Є. Величко// Вісник Житомирського державного університету. – Випуск 4 (76). Педагогічні науки. – Режим доступу: <https://visnyk.zu.edu.ua/Articles/76/12.pdf>
5. Игровая индустрия в 2017 году: тренды, ожидания, советы [Електронний ресурс]. – Режим доступу: <http://renatus.com/ru/2017-gaming-industry-trends>
6. Комп'ютерні ігри як мистецтво [Електронний ресурс]. – Режим доступу: <https://goo.gl/MCXE7G>
7. Ігрова індустрія: тренди та прогнози [Електронний ресурс]. – Режим доступу: <http://innovations.com.ua/ua/articles/op-manage/18565/igrova-industriya-trendi-ta-prognozi>
8. Scratch : Programming for All / Mitchel Resnick, John Maloney, Andres Monroy Hernandez, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, Yasmin Kafai // Communications of the ACM. – 2009. – Vol. 52, No. 11. – P. 60 – 67.
9. Вивчаємо Scratch разом [Електронний ресурс]. – Режим доступу: <https://sites.google.com/comp-sc.if.ua/scratch-together>

УДК 004.413 (045)

Жигаревич О.К., Медведєв М.В.

Луцький національний технічний університет

ІНФОРМАЦІЙНА СИСТЕМА “СТУДЕНТ – ФКНІТ” ЗАСОБАМИ PHP

Жигаревич О.К., Медведєв М.В. Інформаційна ситема “Студент - ФКНІТ” засобами PHP. Розроблено інформаційну систему зі зручним та інтуїтивно зрозумілим інтерфейсом для факультету КНІТ. Окрім інформаційного наповнення система містить модуль додавання, редагування та пошуку потрібної інформації.

Ключові слова: Web-додаток, HTML, CSS, PHP, MySQL, JavaScript, JQuery, інформаційна система.

Жигаревич О.К., Медведєв М.В. Информационная ситема "Студент - ФКНІТ" средствами PHP. Разработана информационная система с удобным и интуитивно понятным интерфейсом для факультета КНІТ. Кроме информационного наполнения система содержит модуль добавления, редактирования и поиска нужной информации.

Ключевые слова: Web-приложение, HTML, CSS, PHP, MySQL, JavaScript, JQuery, информационная система.

Zhigarevich O.K, Medvedev M.V. Informational system "Student - FKNIT" using PHP. An information system was developed with a convenient and intuitive interface for the Department of CNIT. In addition to content, the system contains a module for adding, editing and searching for the necessary information

Keywords: Web-application, HTML, CSS, PHP, MySQL, JavaScript, JQuery, information system.

Постановка проблеми. В даний час з шаленим розвитком ІТ – індустрії в Україні збільшує попит на кваліфікованих спеціалістів в даній сфері. Дана система може використовуватись не тільки студентами факультета КНІТ, але й іншими факультетами, а також факультетами інших університетів. Тому розробка даної інформаційної системи є досить актуальною.

Аналіз існуючих форма та видів інформаційних систем показує, що вони є досить різноманітні – у вигляді Web-додатку (HTML, CSS, PHP, JavaScript), програмного продукту (C++, C#, Java, Python). Прийняття правильного рішення про вибір середовищ і технологій розробки та виду самої системи залежить від обраної аудиторії користувачів.

Мета роботи полягає у розробці інформаційної системи зі зручним та інтуїтивно зрозумілим інтерфейсом, який окрім інформаційного наповнення системи містить модуль додавання, редагування та пошуку потрібної інформації.

Виклад основного матеріалу роботи.

Перш ніж розробляти інформаційну систему, необхідно визначити яким технічним новинкам віддати перевагу. При цьому потрібно врахувати можливу швидкість з'єднання у потенційних користувачів. Тому для створення інформаційної системи було обрано PHP, база даних MySQL, JavaScript, HTML та CSS.

PHP є мовою програмування, код якої можна напяму вставляти в HTML-код сторінок, які у свою чергу будуть оброблені PHP-інтерпретатором. Обробник PHP шукає і обробляє код після знайденого ним парному тегу (<?php та ?>)^[1].

Для CSS використовую тег (<link>). Програми на JavaScript вставляється в будь яке місце HTML за допомогою парного тегу (<script> та </script>)^[2]. У даному проєкті підгрузка php-файлів відбувається динамічно. Для цього було підключенно бібліотеку JQuery.

```
<script src="https://code.jquery.com/jquery-1.12.4.js" integrity="sha256-  
Qw82+bXyGq6MydymqBxNPYTauXXq7c8v3CwiYwLLNXU="
```

```
crossorigin="anonymous"></script>
```

За допомогою вбудованої функції JQuery “\$” ми витягуєм дані з <input> і передаємо зчитуванні дані POST – запитом у php-файл:

```
$(document).ready(function(){  
    $("#submit").click(function(){  
        var name = $("#name").val(),  
            surname = $("#surname").val(),  
            secondname = $("#secondname").val(),
```

```
date = $("#date").val(),
gradebook = $('#gradebook').val(),
group = $("#group").val();

$.post('/loader_st', {name:name, surname:surname,
secondname:secondname, date:date, gradebook:gradebook,
group:group}, function(data){
    $("#div#space").empty();
    $("#space").append(data);
});
});
```

Отриманий результат також зчитується POST-запросом з php-файлу і виводиться на сторінці функцією:

```
$(document).ready(function(){
    $.post('/loader_st', function(data){
        $("#space").append(data);
    });
});
```

Для доступу до адмін-панелі адміністратору системи потрібно спочатку пройти авторизацію. Після входу адміністратору надається доступ для введення змін в такі пункти головного меню як: Додати групу, Додати студенти, Додати новину. При натисненні на пункт меню "Додати групу" адміністратора перекидає на сторінку з формою додавання групи. Після додавання групи, система автоматично переводить адміністратора на сторінку групи з відповідною для групи формою додавання студента. Після додавання студента у правій частині сайту динамічно оновлюється список студентів даної групи з відповідними для кожного студента кнопками: Редагувати та Видалити. Після додавання студента йому також автоматично генерується логін і пароль для авторизації в даній системі, які також виводяться в правій частині сайту під відповідним їм студентам (рисунк 1).

Рисунок 1. Меню додавання студента

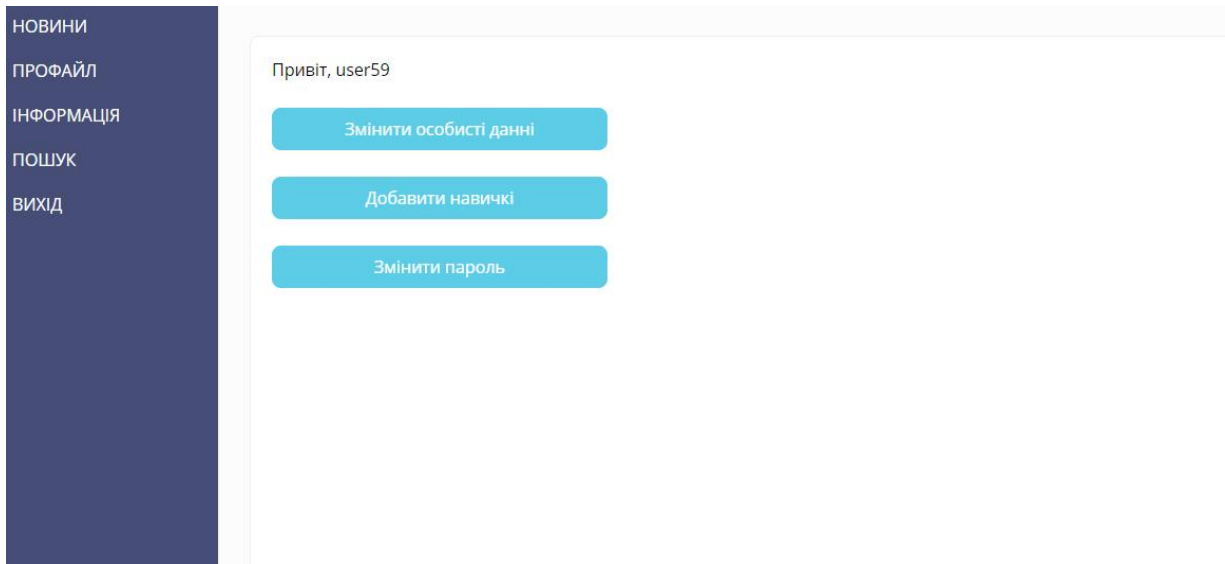


Рисунок 2. Вигляд сторінки студента

Додавання та редагування новин здійснюється адміністратором. При переході у розділ меню “Додати новину” нас автоматично переводить на сторінку з відповідною формою (рисунок 3).

Рисунок 3. Форма додавання новини.

Після додавання новини адміністратора автоматично перекидає на сторінку з опублікованими новинами. Де адміністратору доступні кнопки редагування і видалення новини які розміщені під кожною новиною (рисунок 4).

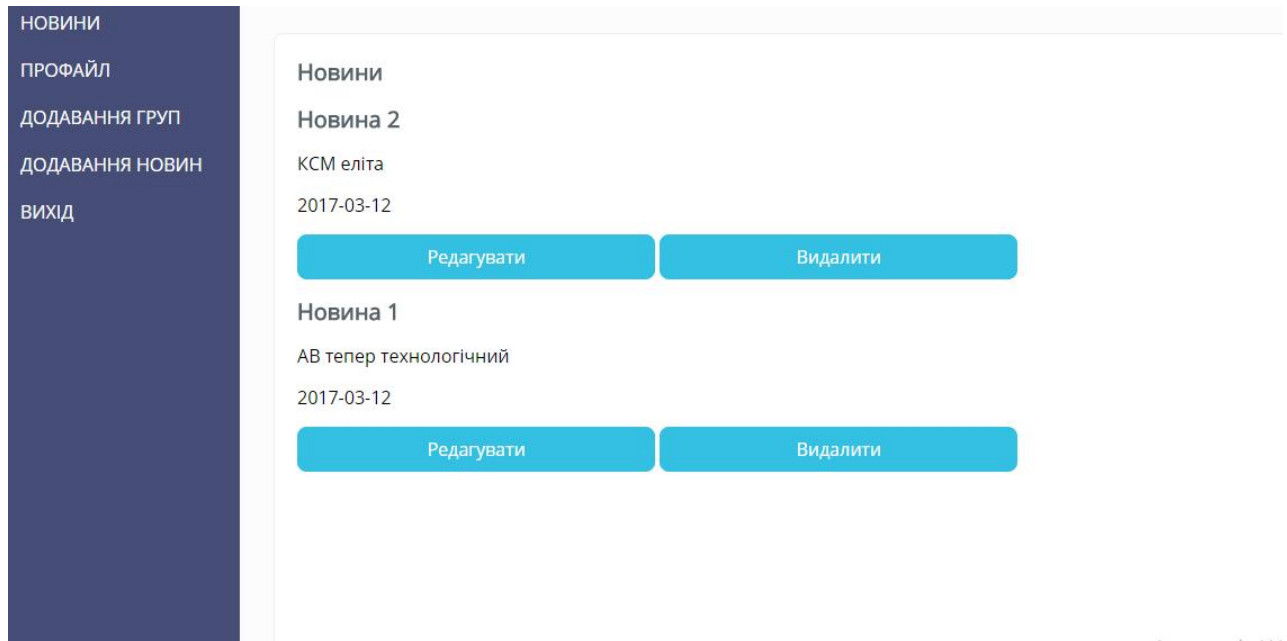


Рисунок 4. Вигляд новин адміністратору.

На рисунку 5 представлено діаграму прецендентів (use case diagram), де можна побачити множину акторів, прецендентів (варіантів використання) обмеженою границею системи, асоціацій між акторами та прецендентами, відношення серед прецендентів та відношення узагальнення між акторами. Діаграма відображає елементи моделі варіантів використання^[3].

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання використовують для описання послуг, які система надає актору^[4].

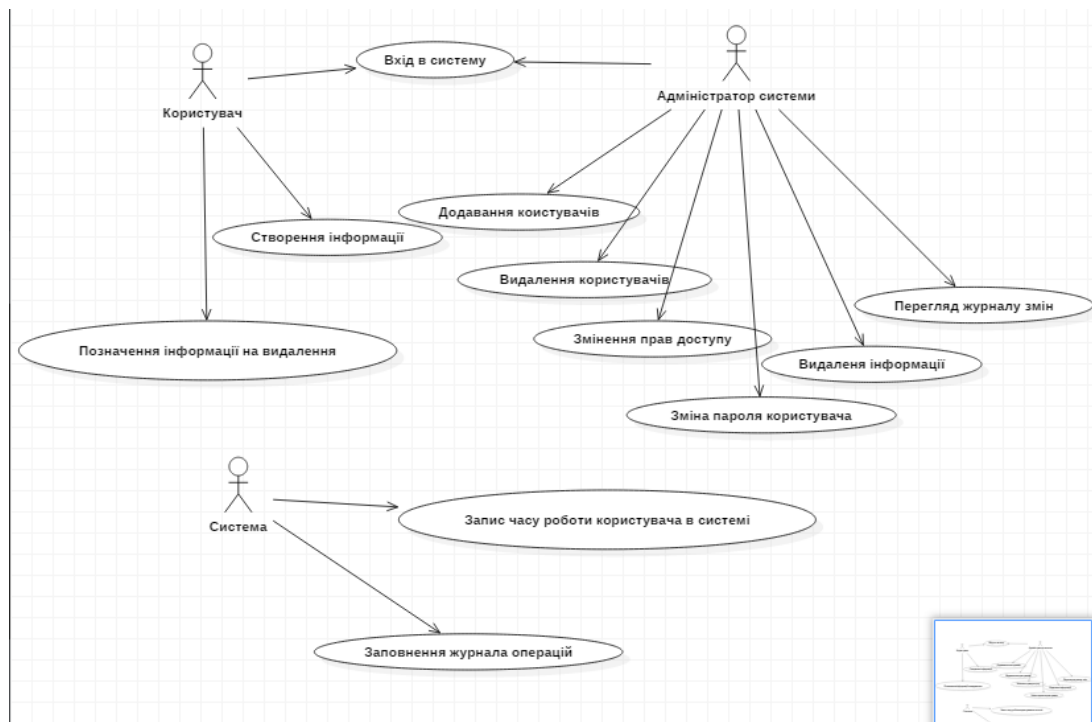


Рисунок 5. Діаграма прецендентів

Висновок. Розроблено інформаційну систему для факультету КНІТ з метою спрощення пошуку студентів HR-менеджерами ІТ-компаній. Даний програмний продукт має змогу збільшити працевлаштування студентів. Система має модуль додавання, редагування та пошуку певної інформації.

Розроблений програмний продукт засобами PHP. В якості локального веб-сервера для реалізації зв'язку "клієнт-сервер" використано Open Server.

1. PHP.SU: [Електроний ресурс] // Основи PHP – Спільнота forum.php.su. – Режим доступу: <http://www.php.su/php/?php/>
2. Learn.javascript.ru: [Електроний ресурс] // Основи JavaScript – Режим доступу: <https://learn.javascript.ru/hello-world>
3. [Booch,Rumbaugh,Jacobson] Гради Буч, Джеймс Рамбо, Ивар Джекобсон – Язык UML. Руководство пользователя. Издательство ДМК Пресс, 2007 г., 496 с. Классика от создателей UML.
4. [Larman] Крэг Ларман – Применение UML 2.0 и шаблонов проектирования. Издательство Вильямс, 2008 г., 736 с. Отражает доминирующий сегодня инженерный подход к ОО АП.

УДК 514.181.6 + 514.182

Журило А. Г., к.т.н., доц. Сівак Є. М., к.т.н.

Національний технічний університет «Харківський політехнічний інститут»

ДЕЯКІ ПИТАННЯ ЩОДО КРЕСЛЕННЯ КІЛ ПРИ ПОБУДОВІ АКСОНОМЕТРИЧНИХ ПРОЕКЦІЙ

Журило А. Г., Сівак Є. М. Деякі питання щодо креслення кіл при побудові аксонометричних проекцій. У статті розглянуті спрощення при кресленні кіл у аксонометричних проекціях. Показано практичне застосування зазначених спрощень. Наведено приклади використання спрощень при кресленні кіл на практиці.

Ключові слова: аксонометрія, ДСТУ, коло, діаметр, ромб, практичне застосування.

Журило А. Г., Сивак Е. М. Некоторые вопросы построения кругов при выполнении аксонометрических проекций. В статье рассмотрены упрощения изображения кругов при вычерчивании аксонометрических проекций. Показано практическое применение указанных упрощений. Приведены примеры вычерчивания кругов и окружностей на практике.

Ключевые слова: аксонометрия, ГОСТ, круг, диаметр, ромб, практическое применение.

Zhurilo A. G., Sivak E. M. Some questions of construction of circles while performing the axonometric projection. The article discusses the simplification of the image of circles when drawing the axonometric projection. The practical application of these simplifications. Examples of drawing circles and circles in practice.

Keywords: axonometry, ISO, circle, diameter, rhombus, 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 проекції 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 проекцій, 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 проекції, не передбачені ГОСТ 2.317 – 69 або ДСТУ ISO 5456-3:2006 [2, 3].

Невирішені частини проблеми. Метою статті є доведення значного практичного значення креслення кіл при виконанні аксонометричних проекцій [7, 10].

Мета дослідження. Чималі труднощі викликає побудова кіл при кресленні аксонометричних проекцій. Справа в тому, що найбільш часто коло в аксонометричній проекції зображують еліпсом. А його побудова – непроста справа.

Якщо розглядати еліпс як паралельну проекцію кола, то можна вважати сполучені діаметри еліпса проекціями двох взаємно перпендикулярних діаметрів зображуваного кола. Цей випадок задання еліпса часто зустрічається в практиці складання креслеників.

На даних сполучених діаметрах еліпса (MN і KL) будують паралелограм (рис. 1), проводячи через кінці кожного діаметра прямі, паралельні іншому діаметру. Потім поділяють на однакову кількість рівних частин один з діаметрів еліпса і сторони паралелограма, що є паралельними іншому діаметру, нумеруючи точки розподілу, як показано на рис. 1. Проводячи з точок K і L прямі через точки розподілу, одержують у перетині однойменних (з однаковими номерами)

прямих точки еліпса. У побудованому таким способом еліпсі не намічені його осі, тоді як вони часто бувають необхідні.

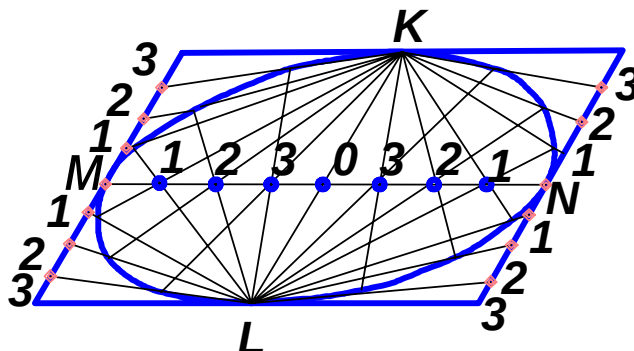


Рис. 1. Побудова еліпса за його сполученими діаметрами

Якщо еліпс побудований і потрібно позначити його осі, треба перетнути еліпс колом довільного радіуса, описаного з центра O (рис. 2). З'єднавши точки 1 і 2 перетину кола з еліпсом хордою 1–2, проводять велику вісь AB паралельно хорді 1–2, а малу вісь CD – перпендикулярно 1–2.

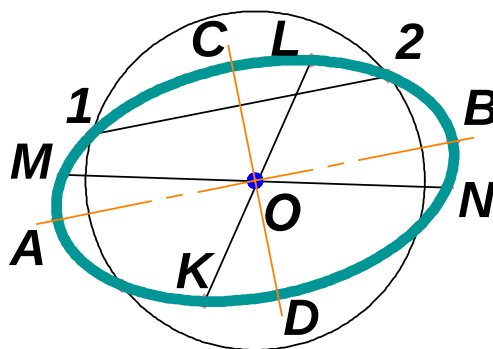


Рис. 2. Побудова еліпса за його сполученими діаметрами

Цей спосіб оснований на перетворенні кола в еліпс і був запропонований Ритцем у 1845 р. Ще більш спростити його дотепер не вдалося. За заданими сполученими діаметрами MN і KL креслять осі еліпса, після чого можна побудувати і сам еліпс будь-яким з наведених вище способів побудови еліпса за його осями, див. рис. 3.

Щоб побудувати осі еліпса, через центр O проводять пряму ON_1 , перпендикулярну до MN , і відкладають на ній від точки O відрізок $ON_1 = ON = OM$. Через точку N_1 та кінець L іншого діаметра проводять пряму і поділяють відрізок N_1L навпіл (точка O_1 – середина відрізка N_1L). Приймаючи точку O_1 за центр, описують коло радіусом OO_1 . Це коло перетинає пряму N_1L в точках G і H , через які проходять осі еліпса. Відрізок N_1H дорівнює великій півосі еліпса, а відрізок N_1G – малій півосі.

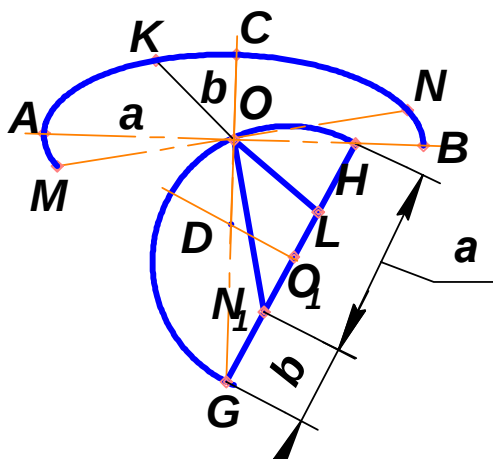


Рис. 3. Побудова еліпса за методом Ритца

Розглянуті способи варто застосовувати в тих випадках, коли еліпс має великі розміри і потребує побудови досить великої кількості його точок (наприклад, більше восьми) або коли потрібно одержати додаткові точки еліпса на окремих його ділянках для більш точного проведення кривої, що використовується в розв'язанні задачі.

У тих випадках, коли еліпс на кресленку невеликий і для його обведення не потрібно багато точок, можна користуватися спрощеними способами, які наведені нижче.

На заданих сполучених діаметрах 1–3 та 2–4 (у розглянутому випадку вони дорівнюють один одному) будують паралелограм (рис. 4, а), проводячи через кінці кожного діаметра сторони паралелограма, паралельні іншому діаметру. Потім з точок 1 та 3 проводять прямі $A1$ і $A3$ під кутом в 45° до напівсторони $1C$ і відзначають точку A перетину цих прямих. З точки 1 як з центра описують дугу кола радіуса $A1$. Через точку B перетину дуги кола зі стороною $1C$ проводять пряму $B6$ паралельно 1–3 та відзначають точки 5 і 6 перетину прямої $B6$ з діагоналями паралелограма. Точки 5 і 6 належать еліпсу.

Провівши далі прямі 5–8 і 6–7, паралельні діаметру 2–4, одержують у перетині цих прямих з діагоналями паралелограма точки 7 та 8, що також належать еліпсу. Таким чином, усього отримують вісім точок еліпса: точки 1, 2, 3 і 4 – кінці сполучених діаметрів еліпса і точки 5, 6, 7, 8, що побудовані описаним способом. Цих восьми точок цілком достатньо, щоб накреслити еліпс, якщо він невеликий за розмірами.

Зазначимо, що в наведеному на рис. 4, а випадку осі еліпса збіглися з діагоналями паралелограма, тому що сполучені діаметри рівні між собою і паралелограм перетворився у ромб.

У загальному випадку задання сполучених діаметрів осі еліпса не збігаються з діагоналями паралелограма. Тоді побудова восьми точок еліпса не зміниться, а осі еліпса можна визначити так, як це було зроблено на рис. 2.

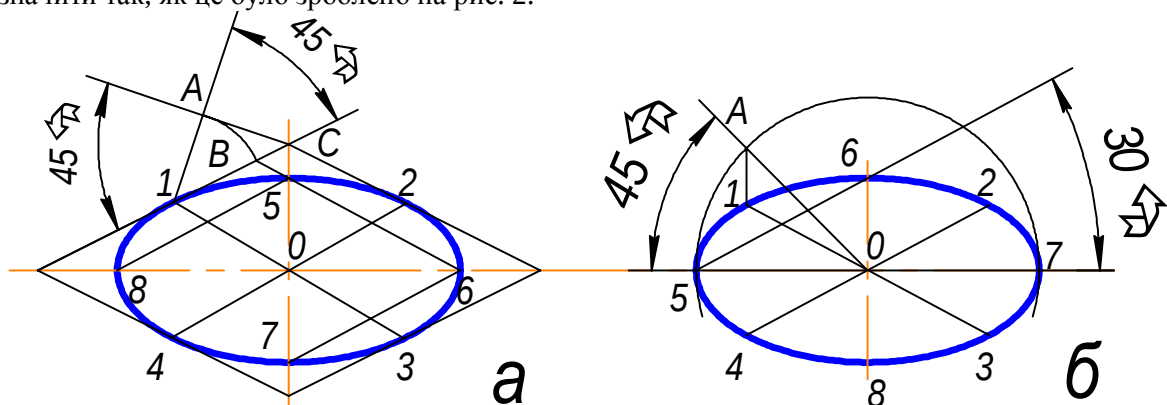


Рис. 4. Побудова еліпса спрощеними способами

Цей спосіб (рис. 4, б) визначення величини осей еліпса і побудови восьми його точок за двома сполученими діаметрами, запропонований Ю. Порсінім, вимагає меншого числа побудов і більш точний, ніж спосіб 6 (рис. 4, а). Але він придатний лише для побудови прямокутної ізометрії кола, що розташоване в кожній із площин координат або їй паралельній, і для побудови еліпса у горизонтальній ізометрії, тоді як спосіб 6 придатний для усіх випадків задання еліпса його сполученими діаметрами.

Розглянемо сутність запропонованого способу на прикладі побудови еліпса, що являє собою прямокутну ізометрію окружності, розташованої в площині Π_1 (XOY) або їй паралельній.

Задано сполучені діаметри еліпса 1–3 та 2–4, які є паралельними аксонометричним осям OY' та OX' . Проводимо через центр еліпса O пряму AO під кутом в 45° по напрямку великої осі еліпса. Через точку 1 проводимо пряму $A1$ перпендикулярно великій осі еліпса і позначаємо точку A перетину прямих AO і $A1$. З центру O радіусом AO описуємо коло до перетину з великою віссю в точках 5 та 7, які і є кінцями великої осі еліпса.

Через точку 5 проводимо пряму 5–6 під кутом 30° до великої осі еліпса (або, що те ж саме, паралельно діаметру 2–4) і позначаємо точку 6 перетину цієї прямої з вертикальною прямою 06 , що є перпендикулярною до великої осі еліпса. Точка 6 – кінець малої осі еліпса. Другий кінець малої осі – точку 8 – можна отримати, відклавши від центра O вниз відрізок $O8 = O6$ або

провівши через точку 7 пряму 7–8, паралельну прямій 2–4.

Визначивши вісім побудованих точок, можна достатньо точно викреслити еліпс. Окрім того, тут визначаються величини осей еліпса, тобто, можна одержати будь-яку його точку одним зі способів побудови еліпса по його осях, а також робити інші побудови, основані на перетворенні кола в еліпс.

Звернемо увагу ще на одну важливу залежність: відношення відрізків $O1/AO = 0,82$. Це графічний спосіб взаємного переходу натуральних координат в аксонометричні для прямокутної ізометрії.

На заданих сполучених діаметрах AB і CD будують паралелограм 1–2–3–4, провівши через кінці кожного діаметра його сторони (рис. 5), паралельні іншому діаметру. Потім через кінець B одного з діаметрів проводять пряму $B5$, перпендикулярну до цього діаметра, і відкладають на ній відрізок $5B = OB$.

З центру O радіусом OB описують коло, що перетинає продовження діаметра AB в точках 6 та 7. Далі будують новий паралелограм 6–8–7–9, проводячи його сторони паралельно діагоналям першого паралелограма. Точки K, L, M, N перетину сторін нового паралелограма з діагоналями першого паралелограма є точками еліпса.

Осі еліпса, якщо вони необхідні, можна позначити за способом, що наведений на рис. 2.

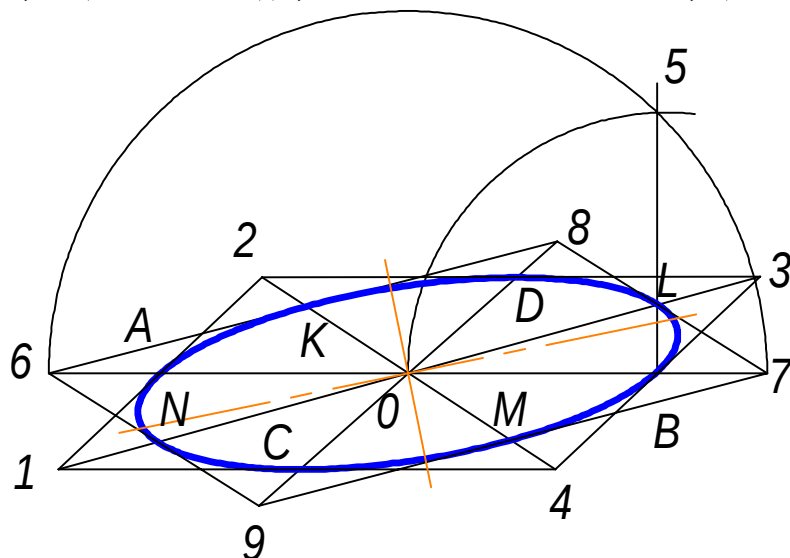


Рис. 5. Побудова еліпса, вписаного в паралелограм

У деяких випадках у практиці аксонометричних проєкцій еліпси заміняють овалами — кривими, що складені з дуг окружностей, тобто так званими коробовими кривими. Побудову овалів можна знайти у всіх підручниках і у всіх довідниках з креслення.

Не заперечуючи відомих рекомендацій із заміни еліпсів овалами, необхідно визначити, що побудова овала нітрохи не простіша побудови еліпса, наприклад по восьми точках.

У той же час овал, навіть близький за своїм обрисом до еліпса, у змісті наочності зображення часто сильно поступається останньому. Якщо ж потрібно розв'язати на аксонометричному кресленнику якусь задачу з використанням проєкції кола, то заміна еліпса овалом звичайно стає неприпустимою.

Таким чином, заміна еліпсів овалами часто переваг не дає і може бути виправдана лише в окремих випадках, пов'язаних з навчанням побудови аксонометричних проєкцій і тим, що овал обводиться циркулем, а не по лекалу, що істотно полегшує їхнє креслення, особливо для початківців.

Овал зручно будувати, вписуючи в ромб, що є ізометричною проєкцією квадрата. Побудова овалу, вписаного в ромб, виконується у такій послідовності.

Спочатку будують ромб зі стороною, що дорівнює діаметру зображуваного кола (рис. 6, а). Для цього через точку O проводять ізометричні осі X' та Y' . На них від точки O відкладають відрізки, рівні радіусу кола, що зображують. Через точки a, b, c та d проводимо прямі, паралельні осям, та отримуємо ромб.

Велика вісь овала розташовується на великій діагоналі ромба.

Далі вписують у ромб овал. Для цього з вершин тупих кутів (точок A та B) описують дуги. Їх радіус R дорівнює відстані від вершини тупого кута (точок A та B) до точок c, d або a, b відповідно (рис. 6, б).

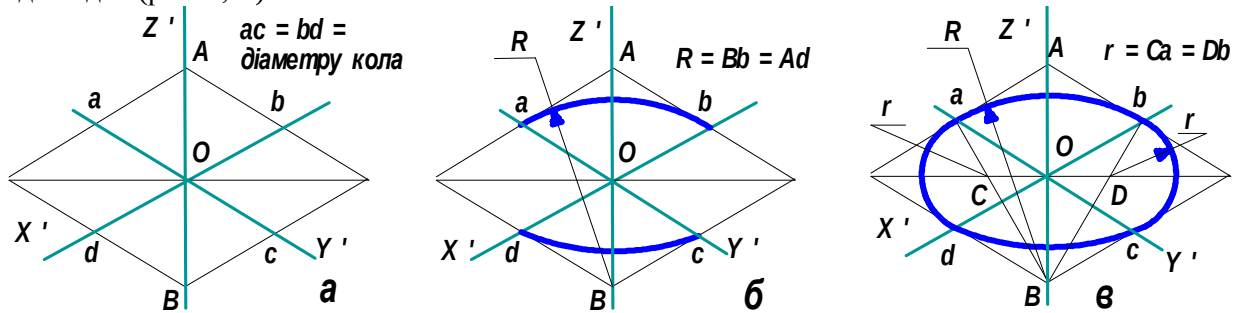


Рис. 6. Побудова овалу, вписаного в ромб

Через точки B та a, B і b проводять прямі. В перетині прямих Ba та Bb з більшою діагоналлю ромба знаходять точки C і D (рис. 6, в). Ці точки будуть центрами малих дуг. Їх радіус r дорівнює Ca (або Db). Дугами цього радіуса плавно з'єднують великі дуги овала.

Ми розглянули побудову овала, що розташований у площині Π_1 , яка перпендикулярна осі z на ортогональному кресленнику. Овали, що знаходяться в площині Π_2 (і перпендикулярні осі y), та в площині Π_3 (і перпендикулярні осі x), будують так само. Тільки для овалу, перпендикулярного осі y , побудову здійснюють на осях x та z , а для овала, перпендикулярного осі x , — на осях y і z .

Висновки та перспективи подальшого дослідження. Розглянуто деякі способи побудови еліпсів та овалів, які можуть застосовуватися при виконанні наочних креслеників без втрат для їхньої ясності.

Наочні аксонометричні зображення можуть варіюватися в дуже широких межах: від ілюзорного зображення до схематичного креслення. Тому застосування побудови еліпсів та овалів 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 «ХП». — 2010. — 196 с.
8. Журило А. Г. Основна теорема аксонометрії – теорема Польке-Шварца та її практичне використання / А. Г. Журило, Є. М. Сівак, 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.

УДК 629.113(071):004.01:004.04

Каганюк О. К., к.т.н, доцент, Сітовський В.О., магістрант
Луцький національний технічний університет

РОЗРОБКА КРОС-ПЛАТФОРМНИХ ДОДАТКІВ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ

Каганюк О.К., Сітовський В.О. Розробка крос-платформних додатків для мобільних пристроїв. В даній статті розглядаються питання що до розробки програмного забезпечення для мобільних пристроїв. Розглянуті головні аспекти розробки програмного забезпечення для усіх відомих мобільних платформ, на прикладі ОС Android.

Ключові слова: Builder, крос-платформність, C++, Android.

Каганюк А.К., Ситовский В.А. Разработка кросс-платформных дополнений к мобильным устройствам. В данной статье рассматриваются вопросы разработки программного обеспечения для мобильных устройств. Рассмотрены основные аспекты разработки программного обеспечения для всех известных мобильных платформ, на примере ОС Android.

Ключевые слова. Builder, кроссплатформенность, C ++, Android.

Kaganyuk A.K., Sitovskiy V.A. Development of cross-platform additions to mobile devices. This article discusses the development of software for mobile devices. The main aspects of software development for all known mobile platforms are considered, for example, the Android OS.

Keywords. Builder, cross-platform, C ++, Android.

Однією з основних платформ, яка використовується в мобільних телефонах, є система Андроїд. Дана система характерна відкритим кодом ядра, та своїм розповсюдженням на безкоштовній основі та можливістю реалізації на різноманітних апаратних платформах для різних пристроїв. Дана операційна система охоплює дуже широкий сектор ринкових послуг, що дає можливість затвердити **актуальність** даного напрямку досліджень.

На даний момент програмне забезпечення для різноманітних пристроїв виросло у дуже різноманітні програмні комплекси. Перед початком розроблення будь якого додатку потрібно спочатку провести аналіз доцільності в виборі структури майбутньої програми.

Компанії зі сфери розробки програмного забезпечення пропонують безліч готових програмних продуктів – додатків, для мобільних пристроїв так званих засобів для розв'язання тих чи інших проблем за допомогою потужності мобільних пристроїв та програмного коду самої програми.

Проте часто виникає **проблема** в розв'язуванні нетривіальної задачі, яка потребує не мало зусиль, або просто створення нового програмного продукту. У такому разі єдине, що залишається - це розробка унікального програмного забезпечення. Для того, щоб воно виявилось успішним, воно повинно відповідати усім поставленим вимогам та найвищим стандартам у своїй галузі. Тому кожний створений продукт повинен бути потрібний користувачам, добре продуманий, спроектований та реалізований. Якщо є необхідність поширення програмного додатку серед користувачів, він повинен бути також професійно представлений на ринку.

Головне питання усіх сучасних програм це їхня оптимізація при запуску та прискорення їхньої роботи. Практично усі додатки на сьогодні містять у собі десятки тисяч строк програмного коду і тут постає **проблема** в оптимізації даного коду та структури програми для пришвидшення її роботи та покращення читабельності самого програмного коду для розробників. Це дозволяє зменшити час компіляції та оптимізувати роботу готового кінцевого продукту.

Взагалі розробка будь-якого програмного продукту потребує великих затрат: як часу, так і ресурсів (людських та матеріальних). Тому під час розробки чогось нового, потрібно врахувати усі ризики, які можуть виникнути під час створення нового програмного продукту.

Впродовж останніх декількох років у розробників все частіше поставало питання реалізації програмних рішень для усіх відомих платформ. Також сюди входять такі **проблеми** як:

1. неприпустимі витрати часу та коштів на розробку;
2. неповноцінність функціоналу;
3. повільна швидкодія.

Сьогодні складно здивувати сучасну людину смартфоном, планшетом, або комп'ютером. Добре це чи погано, але ці пристрої стали частиною нашого життя, і вони продовжуватимуть

впливати на масову культуру. Більш того, їх розробкою займаються декілька провідних конкуруючих компаній. З точки зору розробника, таке різноманіття стає швидше перешкодою, адже практично для кожної із усіх відомих платформ існує свій набір програмного забезпечення (SDK), що, як правило, не сумісний з іншими платформами. Таким чином, розробка для кожної із платформ окремо, – дороге і складне завдання. Розробнику, який візьметься за створення додатку для Android, IOS та Windows, необхідно володіти такими мовами програмування як Java, Objective-C (або Swift) та C#. У такому випадку одна задача породжуватиме ще цілих три окремих продукти, які буде необхідно розробляти і їх підтримувати.

Для вирішення **проблеми** крос-платформності допустима реалізація у вигляді HTML5-додатку, але він буде досить повільним, і не матиме доступу до системних ресурсів. Як наслідок, реалізація за допомогою веб-технологій не є кращим вибором на даний момент. Додаток повинен працювати з різними платформами не тільки в експериментальних умовах, а й на самих звичайних мобільних пристроях, подекуди не самих потужних, і з різними розмірами дисплею[7].

Тому, питанням даної статті, полягає в аналізі та продуктивності читання – запису даних у додатках з різними методами взаємодії

Популярність мобільних пристроїв непинно зростає. З кожним роком їх кількість збільшується. У 82% смартфонів, що було продано у 2016 році встановлено операційну систему Android. При цьому за весь 2016 рік було продано більш ніж 1,5 мільярда Android-пристроїв. З іншого боку, близько 40% корпоративного сегменту США використовує мобільні пристрої під керуванням IOS. А тим часом, Microsoft активно займається розробкою та випуском Windows 10 Mobile. Незважаючи на зміцнення позицій в рейтингу продажів смартфонів, говорити про виняткове успіху китайських компаній ще рано. Проте, всі ці Oppo, Vivo і Huawei розвиваються швидше Apple і Samsung, і їх частка на ринку стрімко збільшується:

Таблиця 1.1 – Рейтинг продажів смартфонів 2015 - 2016. Частка ринку

Компанія	4-й квартал 2015	4-й квартал 2016	2015 рік	2016 рік
Samsung	20.2%	17.7%	22.2%	20.8%
Apple	18.6%	17.8%	16.1%	14.5%
Huawei	8.1%	10.2%	7.4%	9.3%
Oppo	3.7%	6.7%	2.8%	5.7%
Vivo	3.3%	5.8%	2.7%	4.8%
Інші	46.2%	41.7%	48.8%	44.9%

Самсунг дійсно втрачає темп продажів, але все одно залишається найуспішнішою компанією в світі. Правда, річні продажі бренду знизилися з 319,7 до 309,4 мільйона смартфонів.

Лідер в четвертому кварталі 2016 року корпорація Apple за підсумками року продемонструвала більше втрат, ніж Samsung. Продажі яблучних телефонів впали з 231,5 до 215,4 мільйона смартфонів.

Корпорація Huawei збільшила тиражі з 107,1 до 138,8 мільйона смартфонів, Oppo підняла продажі - з 39,7 до 84,6 мільйона, а бренд Vivo - з 39,0 до 71,9 мільйона.

За річним накладом видно, що ситуація з Apple розвивається зовсім не так, як хотіли б «чарівники з Купертино». Samsung здригнувся, але встояв. Зараз корейці випускають і продають більше смартфонів, ніж всі три «китайських тигра» разом узяті. При цьому неможливо не помітити і зростання продажів Huawei з фантастичною дворазовою динамікою Oppo і Vivo[1].

При такому різноманітті та розповсюдженні мобільних пристроїв питання крос-платформності стає саме по собі. В цьому спектрі найбільш помітним стає C++Builder 10.1 Berlin, – середовище для крос-платформної розробки мобільних додатків із використанням C#.

Застосування. Огляд основних характеристик та можливостей розробки крос-платформних додатків із використанням C++Builder 10.1 Berlin та C#, враховуючи основні переваги та недоліки.

C++Builder Overview

C++Builder надає можливості для розробки додатків для різних платформ з швидким, сучасним, поліпшеним Clang компілятором C++. Просто компілюємо і відкриваємо додаток на різних платформах і пристроях. Проектуємо призначений для користувача інтерфейс один раз, а потім швидко оптимізуємо для кожної платформи або пристрою - подібно породженню підкласів в коді. C++Builder використовує надійні і потужні засоби для роботи з базами даних рівня підприємства, REST, хмарними сховищами, IoT і іншими. C++Builder прискорює розробку:

приділяючи більше часу кодуванню, менше часу на дизайн користувацького інтерфейсу, не затрачаючи ні хвилини - на портування конкретних платформ.

C ++ Builder надає швидкий, потужний, сучасний C ++; єдиний компілятор, єдиний відладчик, єдину IDE, прекрасні бібліотеки на чотири платформи і дозволяє прискорити розробку, зосередитися на своєму коді і вивести додаток на ринок швидше, ніж з будь-яким іншим інструментарієм C ++.

Новітній C ++: Посилений Clang компілятор C ++

C ++ Builder включає в себе вдосконалений компілятор на основі Clang для Windows (32 і 64), IOS, і Android. Застосовуйте цей компілятор на всіх платформах для швидкої і надійної розробки. Містить значні розширення синтаксису мови для ефективного програмування, які ще не увійшли в стандартний C ++, включаючи властивості і автоматичний підрахунок посилань (ARC), які вам дуже сподобаються. Повна зворотна сумісність для старіших C ++. Він уже має повну підтримку C ++ 11, а підтримка C ++ 17 і Linux очікується найближчим часом.

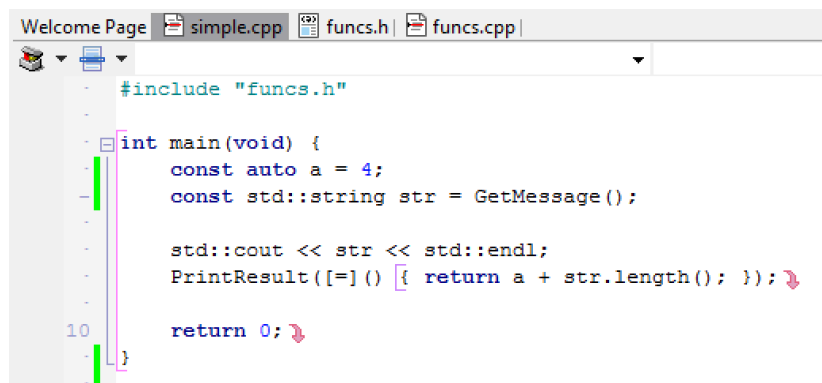


Рис. 1. Програмування за допомогою Clang компілятора C ++

Єдиний компілятор, єдина бібліотека UI (з підтримкою нативних елементів управління), єдина IDE, єдиний набір бібліотек і фреймворків. Повна підтримка чотирьох платформ і це ще не межа.

Набридло переписувати кілька варіантів UI для однієї програми? За допомогою multi-device designer, ви зможете один раз створити основу вашого UI і оптимізувати його для будь-якої платформи або навіть пристрої. Наприклад, створіть мобільний UI, а потім швидко і легко отримаєте UI для OS або iPad з по-справжньому нативним зовнішнім виглядом і поведінкою.

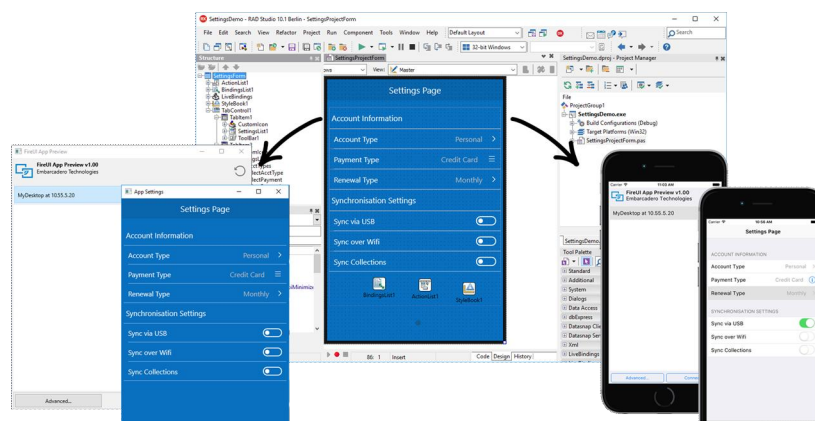


Рис. 2. Фреймворк FireMonkey UI підтримує чотири платформи

Фреймворк FireMonkey UI - за рахунок векторизації і використання GPU - надає незрівняну швидкість роботи UI і якість відтворення. Він навіть підтримує «рідні» елементи управління - наприклад, справжні кнопки в Windows або реальні текстові поля в iOS - а не імітує їх, як роблять інші крос-платформні фреймворки.

Бази даних, REST, web, хмарні сервіси і IoT, бібліотеки згідно з додатками, що працюють на будь-якій платформі і пристрої. Досить возитися з безліччю бібліотек і дизайнерів: вдвічі скоротить час розробки. Використовуйте App Tethering, щоб зв'язати додаток на планшеті або Mac з Windows-версією того ж додатка.

Середовище IDE і вбудований відладчик

Чудове середовище IDE і вбудований відладчик:

1. C ++ автозавершення коду і параметрів (code completion).
2. Перейменування в рефакторингу.
3. Стеження за значеннями при налагодженні у локальних змінних.
4. Інспектор об'єктів (object inspector).
5. Умовні та логічні точки зупинки.
6. Розширювані підказки з поточними значеннями.
7. Вікна налагодження Inspection і evaluation і багато іншого.

Найсудовішим розробникам сподобається надзвичайно потужне вікно асемблерного представлення коду.

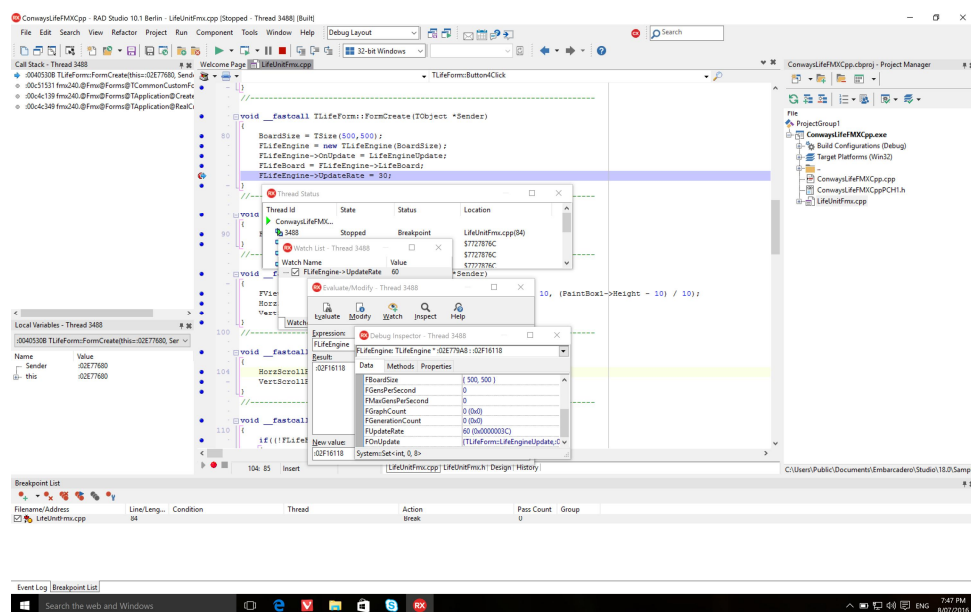


Рис. 3. Середовище розробки RAD Studio 10.1 Berlin

VCL - це всеосяжна, що не має рівних бібліотека підтримки нативних елементів призначеного для користувача інтерфейсу Windows, яка надає можливостей більше, ніж будь-яка інша бібліотека.

Вона повністю нативна, по-справжньому об'єктно-орієнтована, з підтримкою властивостей, подій і візуального редактора з налаштуванням візуальних стилів представлення. Надає величезну кількість елементів управління для будь-якої версії ОС, включаючи Windows 10, і неймовірну підтримку власних елементів, якщо ви вирішите написати свої власні.

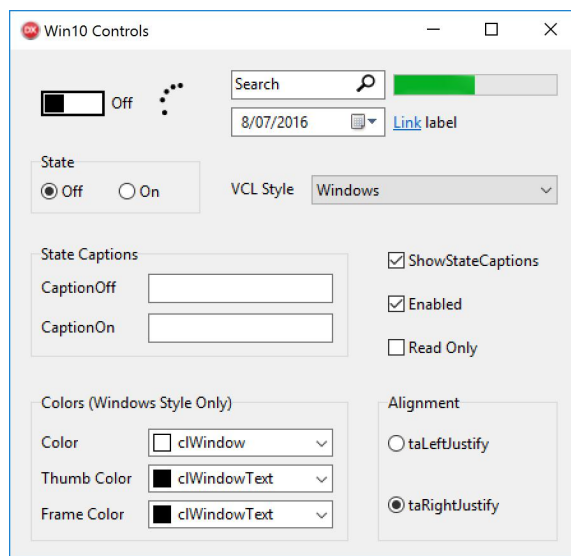


Рис. 4. Налаштуванням візуальних стилів представлення для Windows 10

Швидка розробка для будь-якого масштабу розробки

C ++ Builder дуже популярний у розробників наукових, інженерних та медичних програм, мобільних додатків, корпоративних і інформаційних систем, від легких настільних додатків до гігантських багатоланкових рішень. Він має прекрасні бібліотеки для серйозних розробок, і створює нативні виконувачі модулів. Крім того, що компілятор оптимізує швидкодію додатків, а бібліотеки і фреймворки економлять дорогий час розробки, ви також отримуєте відмінну можливість швидко створювати крос-платформні додатки[3].

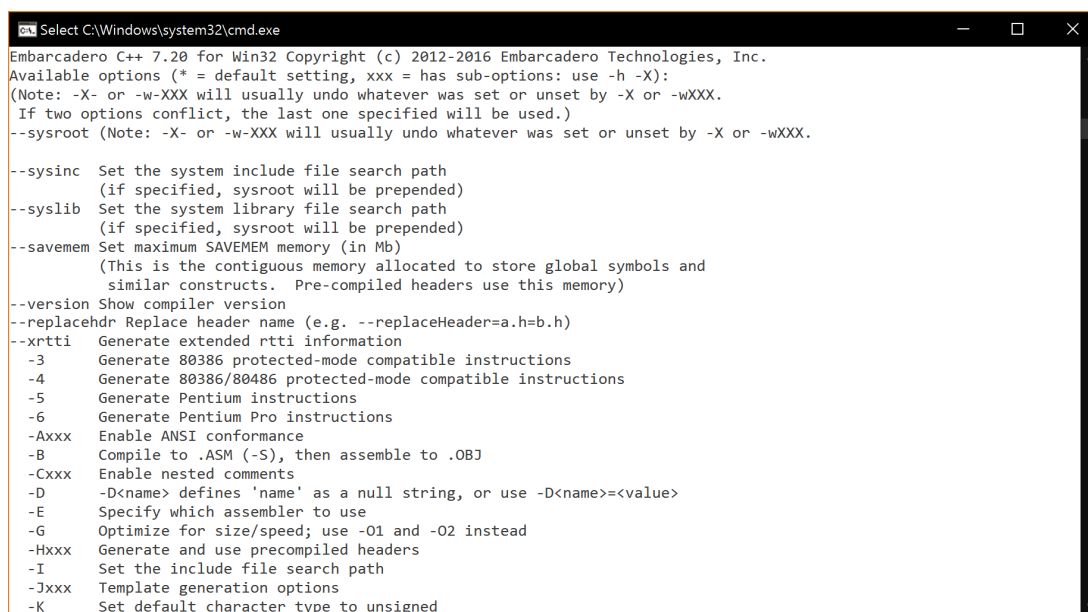


Рис. 5. Компілятор Embarcadero C++ 7.20 for Win32

C ++ Builder 10.1 Berlin – порівняння редакцій

C ++ Builder 10.1 Berlin Architect - найшвидший спосіб створювати і оновлювати інтенсивно працюють з даними, розподілені і сильно взаємодіючі додатки з розвиненим і візуально привабливим для користувача інтерфейсом для Windows 10, Mac, мобільних пристроїв, IoT і інших платформ за допомогою стандартної мови C ++. Ви зможете з легкістю оновлювати VCL- і FMX-додатки для Windows 10, використовуючи нові елементи управління і VCL-стилі для Windows, а також компоненти служб універсальної платформи Windows.

C ++ Builder 10.1 Berlin Enterprise - саме швидкодіюче рішення по розробці програмного забезпечення на C ++ для незалежних і корпоративних розробників. Створюйте клієнт-серверні і багаторівневі взаємодіючі додатки, що підключаються до широкого спектру корпоративних баз даних і хмарних платформ, включаючи Microsoft SQL Server, DB2, Oracle, Sybase, InterBase, Amazon і Microsoft Azure. C ++ Builder Enterprise підтримує всі можливості редакції Professional, а також забезпечує підключення до корпоративних даних за допомогою DataSnap SDK і містить ліцензію розробника на служби Enterprise Mobility Services.

C ++ Builder 10.1 Berlin Professional - найшвидший спосіб створювати і оновлювати інтенсивно працюють розподілені і сильно взаємодіючі додатки з розвиненим і візуально привабливим для користувача інтерфейсом для Windows 10, Mac, мобільних пристроїв, IoT і інших платформ за допомогою стандартної мови C ++. Ви зможете з легкістю оновлювати VCL- і FMX-додатки для Windows 10, використовуючи нові елементи управління і VCL-стилі для Windows, а також компоненти служб універсальної платформи Windows[2].

Таблиця 2. – Порівняння редакцій C ++ Builder:

	Starter	Pro	Enterprise	Architect
Розробка 32-розрядних додатків для Windows	✓	✓	✓	✓
Розробка 64-розрядних додатків для Windows		✓	✓	✓
Розробка універсальних (32- і 64-розрядних) додатків для iOS		Можна придбати додатково	✓	✓
Develop Android apps including Marshmallow		Можна придбати додатково	✓	✓
Розробка додатків для OS X		✓	✓	✓
Develop iOS and Android apps		Можна придбати додатково	✓	✓
Комплексний набір компонентів VCL и FireMonkey	З обмеженнями	✓	✓	✓
Вихідний код бібліотек		✓	✓	✓
Створення додатків для баз даних з локальним і вбудованим підключенням		Можна придбати додатково	✓	✓
Сервіси Enterprise Mobility Services (ліцензія розробника)		Можна придбати додатково	✓	✓
Пакет SDK для багаторівневого програмування DataSnap			✓	✓
Засоби для роботи з базами даних SQL				✓
Засоби моделювання даних				✓
Ліцензія на комерційне використання	З обмеженнями	Повністю	Повністю	Повністю
Доступ до ліцензій на попередні версії		✓	✓	✓

Основні нові можливості RAD Studio 10.1 Berlin (Delphi, C ++ Builder):

- підтримка Android 6.0 (API Level 23);
- поліпшення в Android Service;
- поліпшення в FireDAC для DB2, MongoDB, Firebird, Oracle, PostgreSQL, SQLite;
- новий компонент TAddressBook, який дозволяє отримати доступ до адресної книги на платформі Android і iOS;
- поліпшений Style Designer в FireMonkey;
- поліпшений механізм попереднього перегляду інтерфейсу для різних пристроїв;
- поліпшений механізм підтримки IoT (Інтернет речей);
- новий елемент ListView;

- поліпшений IFMXExtendedClipboardService для роботи з буфером обміну;
- поліпшений редактор списку зображень в компоненті FMX.TImageList;
- підтримка High DPI для Windows-додатків;
- підтримка EMS сервера Apache;
- нові можливості EMS;
- поліпшений BeaconFence для роботи з маячками;
- покращена продуктивність з Bluetooth LE, Beacon;
- підтримка асинхронних HTTP запитів;
- поліпшений клас TStrings, з'явилися нові методи;
- поліпшений TMemIniFile;
- покращена продуктивність з Generics;
- додані залежності в GetIt Package Manager;
- і багато іншого[5].

Команда розробників Embarcadero у першу чергу намагається задовольнити усі потреби та побажання розробників, розуміючи, що саме від них залежатиме, скільки буде розроблено додатків з використанням Builder. Це ми можемо стверджувати, зважаючи на документацію на офіційному сайті, та на кількість доповнень (більша їх частина платна), які можна завантажити та встановити у свій додаток.

Для початківців, які використовують Builder, формат пошуку допомоги на форумах є досить незвичним, – розробник здебільшого буде отримувати відповіді на свої питання на форумах, що присвячено темам, RAD Studio, Delphi, C++Builder, Appmethod and InterBase, Integrated Development Environment (IDE), SQL databases, C++ and Object Pascal programming[4].

Ми виділили основні переваги, які, на нашу думку, є головними у аспектах вирішення поставленого завдання:

- розробка для IOS та MAC, Android, Windows Phone OS з використанням C++ and Object Pascal, Delphi та .Net Framework;
- можливі мови програмування: C++ and Object Pascal, Delphi та можливість програмування на Java, при підключенні спеціальних бібліотек;
- використання рідних UI (User Interface) у додатках для кожної із платформ;
- швидкодія;
- повний доступ до API обраної платформи;
- пробна версія, що містить в собі достатній функціонал для повноцінної розробки;
- підтримка Google Glass, Smart Watch.

Ми вважаємо, що Embarcadero RAD Studio варто використовувати тоді, коли додаток необхідно реалізувати для роботи як мінімум на двох основних платформах, таких як: Android та IOS. Але пристрої, що працюють під їх керуванням різні, починаючи з апаратного забезпечення, закінчуючи логікою роботи операційної системи та ліцензуванням в цілому. C++ Builder надає C++ компілятор C для Windows 10 (32-розрядна та 64-розрядна версія) і мобільних платформ (IOS і Android) з розширеннями RAD для найшвидшої Standard C++ Windows і крос-платформної розробки. Тісна інтеграція з фреймворками VCL для Windows і крос-платформних FMX, підтримка мови C++ 11 і управління пам'яттю для мобільних пристроїв на основі ARC (Автоматичний підрахунок посилань), з зворотною сумісністю. Це оновлення необхідно для розробників C++ Builder для Windows VCL або нового мобільного C++ і розробників настільних комп'ютерів, які перейшли з інших мов і наборів інструментів, таких як Java, Objective C, C#, XCode, Xamarin і Visual Studio[6].

Висновки:

На базі проведеного огляду та аналізу існуючих моделей програмного забезпечення для мобільних пристроїв на базі «Андроїд», можна затвердити наступне:

1. Тема крос-платформності, особливо серед мобільних додатків є досить актуальною;
2. Нами виділена можливість реалізації додатків за допомогою C++ Builder;
3. Embarcadero RAD Studio надає можливість як найшвидшого вирішення поставленої проблеми з як найменшими витратами ресурсів, без втрати швидкодії.

1.Рейтинг продажів смартфонів - підсумки 2016 року [Електронний ресурс] Режим доступу: <https://5nch.com/lidery-prodazh-smartfonov-2016-rejting-i-godovoj-otchot/>

2. C++Builder 10.1 Berlin [Електронний ресурс] Режим доступу: <http://www.interface.ru/home.asp?artId=38525>
3. C++Builder Overview [Електронний ресурс] Режим доступу: <https://www.embarcadero.com/ru/products/cbuilder>
4. Embarcadero Community online technical forums [Електронний ресурс] Режим доступу: <https://community.embarcadero.com/forum>
5. Реліз RAD Studio 10.1 Berlin (Delphi, C ++ Builder) [Електронний ресурс] Режим доступу: <https://sohabr.net/habr/post/282055/>
6. Покращений CLANG C ++ компілятор для Windows 10 і мобільних пристроїв [Електронний ресурс] Режим доступу: <https://www.embarcadero.com/ru/products/rad-studio>
7. Розробка крос-платформних додатків для мобільних пристроїв [Електронний ресурс] Режим доступу: <http://eprints.zu.edu.ua/22019/>
8. Автоматический поиск ошибок синхронизации в приложениях на платформе Android" В.П. Иванников, С.П. Варганов, М.К. Ермаков
9. Разработка приложения для android на java» Елюбаева Ж.Ж. Кандидат технических наук, Евразийский национальный университет имени Л.Н.Гумилева
10. Выработка и анализ требований к защищенной мобильной операционной системе» О.И. Бокова, Д.М. Михайлов, М.И. Фроимсон

УДК 004.65

Коцюба А.Ю., Саковський С.В.

Луцький національний технічний університет

ПРО ІМПОРТУВАННЯ БАЗ ДАНИХ В ANDROID-ДОДАТОК

Коцюба А.Ю., Саковський С.В. Про імпортування баз даних в Android-додаток. Проведено аналіз сучасних підходів щодо зв'язування додатків створених для ОС Android з базами даних під різними СКБД. Описано основні труднощі, які можуть при цьому виникнути. Запропоновано два основні алгоритми вирішення цієї проблеми: паралельний та послідовний. Детальніше розглянуто послідовний алгоритм зв'язування, кінцева мета якого це імпортування БД, створеної, наприклад, на базі СКБД SQLite, в Android-додаток. Запропоновано та проілюстровано за допомогою відповідних листингів коду два різних способи вирішення цієї проблеми.

Ключові слова: імпортування баз даних, Android-додаток, СКБД SQLite.

Коцюба А.Ю., Саковский С.В. Об импортировании баз данных в android-приложение. Проведен анализ современных подходов по связыванию приложений созданных для ОС Android с базами данных под различными СУБД. Описаны основные трудности, которые могут при этом возникнуть. Предложено два основных алгоритма решения этой проблемы: параллельный и последовательный. Подробнее рассмотрено последовательный алгоритм связывания, конечная цель которого это импорт БД, созданной, например, на базе СУБД SQLite, в Android-приложение. Предложены и проиллюстрированы с помощью соответствующих листингов кода два различных способа решения этой проблемы.

Ключевые слова: импортирование баз данных, Android-приложение, СУБД SQLite.

Kotsyuba A.Yu, Sakovskyy S.V. About importing databases into the android-application. The analysis of modern approaches for linking applications created for the Android OS with databases under different DBMSs is carried out. The main difficulties that can arise in this case are described. Two main algorithms for solving this problem are proposed: parallel and sequential. A sequential binding algorithm whose ultimate goal is database import created, for example, based on DBMS SQLite, into the Android application, is considered in detail. Two different ways of solving this problem are proposed and illustrated with the help of the corresponding code listings.

Keywords: importing database, Android-application, DBMS SQLite.

Вступ. Однією із головних проблем програмного забезпечення високо рівня була і досі залишається реалізація взаємодії з незалежним програмним забезпеченням обробки інформації для роботи з великими об'ємами інформації. Рішення цієї проблеми беруть на себе системи керування базами даних (СКБД). Вони реалізують інтерфейс між апаратними засобами накопичення інформації та програмним забезпеченням обробки інформації. При цьому в останні роки широкої популярності набула проблема розробки додатків для мобільних пристроїв (наприклад таких, які працюють на платформі ОС Android). У процесі створення Android-додатків досить часто виникає питання про можливість коректного імпортування баз даних великого об'єму зі складною, наприклад, схемою зв'язків.

Аналіз останніх досліджень та публікацій. Більше дізнатися про створення Android-додатків можна з електронного ресурсу [4]. У джерелах [1, 2, 5] наведено найвідоміші прийоми створення додатків під платформу ОС Android. Використання баз даних та їх співпрацю з такими додатками описано в електронних ресурсах [3, 5]. Для дослідження методик вирішення поставленої в роботі проблеми можна використати джерела [2, 4, 5]. Роботу з базами даних за допомогою СКБД висвітлено працях [3, 5].

Мета роботи – розгляд методик перенесення (або імпортування) бази даних в Android-додатки та визначення найефективнішої з них. Слід зауважити, що мету можна розділити на початкову (яка і є основною) та кінцеву (єдина проблема, яка стоїть на заваді її практичної реалізації – це знаходження найбільш оптимальних та простих способів налаштування ефективної взаємодії Android-додатків з відповідними БД). Кінцевою метою роботи є написання хоча б двох можливих способів такої методики імпортування та підключення готових баз даних до Android-додатків, що є відносно не складними в реалізації та одночасно оптимально ефективними.

Постановка наукової проблеми та обґрунтування вибору способу її рішення. Одним із найпоширеніших методів налаштування взаємодії Android-додатку з коректно створеною БД є паралельне її створення під час написання інтерфейсу та коду додатка. Зазвичай сучасні середовища, в яких реалізовано можливість створення такого додатка дозволяють це робити. Але при цьому є обмеження на СКБД (є ряд систем керування, які запропоновані розробнику); і в процесі ускладнення БД виникає багато проблем, вирішення яких пов'язано зі значними труднощами (при цьому не завжди вдається налаштувати оптимально коректну взаємодію).

Для випадків, коли БД є об'ємною і деякі структурні її характеристики в процесі розробки будуть весь час оптимізуватися, вищеописаний метод паралельного створення зазвичай не дасть

таких результатів як метод послідовного створення. Він полягає у тому, що спочатку БД і Android-додаток створюються незалежно (при написанні додатку, очевидно, потрібно враховувати подальше його зв'язування з БД); а далі сформована БД з вибраної розробником СКБД імпортується в додаток, який "готовий" до такої взаємодії. Виникає питання при виборі останнього підходу: з якими СКБД в такому випадку найефективніше співпрацювати? Дати однозначну відповідь, яка б задовольнила всіх компетентних розробників таких додатків, на нашу думку неможливо, оскільки технології в цьому напрямку весь час змінюються і майже все залежить від навиків та вмінь розробника. Тому нами запропоновано огляд деяких найвідоміших СКБД в контексті проблеми імпортування розробленої БД в Android-додаток.

MySQL – вільна (відкрита) система керування базами даних. MySQL є власністю компанії Oracle Corporation. Цю систему управління базами даних з відкритим кодом було створено як альтернативу комерційним системам. MySQL із самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL – одна з найпоширеніших таких систем. Вона використовується, у першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL є рішенням для малих і середніх додатків. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або віддалені клієнти, проте до дистрибутиву входить бібліотека внутрішнього сервера, що дозволяє включати MySQL до автономних програм. Вихідні коди сервера компілюються на багатьох платформах. Для некомерційного використання MySQL є безкоштовною. Можливості сервера MySQL можна сказати наступне:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн.;
- висока швидкість виконання команд;
- наявність простої та ефективної системи безпеки;
- можливість імпортування БД у різні середовища, зокрема і у андроїд подібні програми.

Але незважаючи на всі плюси даного середовища існують і мінуси, з більш розвиненими функціями у вибраній нами сфері використання.

СКБД **Oracle** – це найпотужніший програмний комплекс, що дозволяє створювати додатки будь-якої складності. Ядром цього комплексу є база даних, що зберігає інформацію, кількість якої за рахунок наданих засобів масштабування практично безмежна. З високою ефективністю працювати з цією інформацією одночасно може практично будь-яка кількість користувачів (за наявності достатніх апаратних ресурсів), не проявляючи тенденції до зниження продуктивності системи при різкому збільшенні їхньої кількості.

Однією складовою успіху СКБД Oracle є те, що вона поставляється практично для всіх існуючих на сьогодні операційних систем. Працюючи під ОС Sun Solaris, Linux, Windows або під іншими платформами з продуктами Oracle не буде виникати ніяких проблем у роботі. Мінусом даного середовища є те, що воно платне, і потребує ліцензії на використання.

SQLite – це відкрита вбудована реляційна система керування базами даних. Основними характеристиками SQLite є:

- високий рівень мобільності (незалежності від платформи);
- простота у використанні;
- ефективність та надійність;
- має досить невеликий розмір (близько 300 кілобайт), що дозволяє використовувати цю СКБД на малопотужних пристроях (наприклад iPhone).

Крім цього SQLite доступний на будь-якому Android-пристрої, його не потрібно встановлювати окремо. SQLite підтримує типи TEXT (аналог String в Java), INTEGER (аналог long в Java) і REAL (аналог double в Java). Решта типів слід конвертувати, перш ніж зберігати в базі даних. SQLite сама по собі не перевіряє типи даних, тому ви можете записати ціле число в колонку, призначену для рядків, і навпаки.

На нашу думку саме СКБД SQLite є найбільш раціональним варіантом вибору середовища для імпортування баз даних в Android-додаток. Крім цього по функціональних можливостях вона не уступає вищеописаним системам керування. Звичайно, існують і інші не менш ефективні рішення даної проблеми. Але метою даної роботи є не лише вибір СКБД такої, що дозволила б задовільнити потреби будь-якого мобільного додатка в контексті описаної проблеми, а і опис процесу імпортування, при якому можна налагодити ефективну взаємодію з розробленим під БД Android-додатком.

Імпортуння за допомогою СКБД SQLite. Зазначимо, що при використанні системи керування SQLite користувачу не потрібно застосовувати можливості віддаленого сервера, база даних після перенесення в додаток зберігається уже на самому мобільному пристрої.

Опишемо два способи такого процесу перенесення в додаток (імпортування).

Спосіб №1. Є багато програм різних програм для роботи з SQLite. Запропонуємо SQLite Database Browser, яка є відкритою і легко інсталується в ПК розробника під ОС Windows.

Спочатку необхідно виконати копіювання заздалегідь підготовленого файлу бази даних в папку **assets**, а далі програмно можна скопіювати даний файл в потрібну системну папку бази даних мобільного додатка. При цьому слід зважати на те, що крім таблиці створеної розробником для додатка, Android також створює для своїх цілей нову таблицю **android_metadata** у вашій базі. Тому при ручному створенні бази вам необхідно створити як мінімум дві таблиці: системну і свою робочу. Для цього відкриємо створену базу і додамо нову таблицю під назвою "android_metadata":

```
CREATE TABLE "android_metadata" ("locale" TEXT DEFAULT 'en_US')
```

Додамо в таблицю один рядок:

```
INSERT INTO "android_metadata" VALUES ('en_US')
```

Далі створимо свої таблиці для роботи. При необхідно перейменувати поле "id" на "_id", як вимагає Android. У SQLite Database Browser це можна зробити, натиснувши на кнопку *Редагувати*, а потім вибрати таблицю, яку потрібно змінити, та поле для перейменування. Після виконання зазначених операцій база даних готова для використання у вашому додатку. Помістимо файл бази даних в папку **assets** мобільного проекту і створимо новий клас, що успадковується від SQLiteOpenHelper. Детальніше код додатку, що відповідає маніпуляціям наведено на наступному лістингу коду для переносу БД в Android-додатки:

```
public class DataBaseHelper extends SQLiteOpenHelper {  
    // Шлях до бази даних вашої програми  
    private static String DB_PATH="/ data / data / YOUR_PACKAGE / databases /";  
    private static String DB_NAME="myDBName";  
    private SQLiteDatabase myDataBase;  
    private final Context mContext;  
  
    // Конструктор  
    // Приймаємо і зберігаємо посилання на переданий контекст для доступу до ресурсів дод.  
    // @param context  
    public DataBaseHelper (Context context) {  
        super (context, DB_NAME, null, 1); this.mContext=context; }  
  
    // Створюємо порожню базу даних і перезаписує її нашої власною базою  
    public void createDataBase() throws IOException {  
        boolean dbExist=checkDataBase ();  
        if (dbExist) { /* Нічого не робимо - база вже є */ }  
        else {  
            // Викликаючи цей метод створюємо порожню базу, пізніше вона буде переписана  
            this.getReadableDatabase ();  
            try {  
                copyDataBase (); }  
            catch (IOException e) {  
                throw new Error ( "Error copying database"); }  
        }  
    }  
  
    // Перевіряємо, чи існує вже ця база, щоб не копіювати її кожен раз при запуску прогр.  
    // @return true якщо існує, false якщо не існує  
    private boolean checkDataBase() {  
        SQLiteDatabase checkDB=null;  
        try {  
            String myPath=DB_PATH+DB_NAME;  
            checkDB=SQLiteDatabase.openDatabase (myPath,null,SQLiteDatabase.OPEN_READONLY); }  
        catch (SQLiteException e) { /* База ще не існує */ }  
        if (checkDB!=null) { checkDB.close(); }  
        return checkDB!=null?true:false; }  
  
    // Копіюємо базу з папки assets замість створеної локальної БД  
    // Виконуємо шляхом копіювання потоку байтів.  
    private void copyDataBase() throws IOException {  
        // Відкриваємо локальну БД як вхідний потік
```

```
InputStream myInput=mContext.getAssets().Open(DB_NAME);
// Шлях до новоствореної БД
String outFileName=DB_PATH+DB_NAME;
// Відкриваємо порожню базу даних як вихідний потік
OutputStream myOutput=new FileOutputStream(outFileName);
// Переміщаємо байти з вхідного файлу в вихідний
byte [] buffer=new byte[1024];
int length;
while ((length=myInput.read (buffer))>0) { myOutput.write(buffer,0,length); }
// Закриваємо потоки
myOutput.flush();
myOutput.close();
myInput.close(); }

public void openDataBase () throws SQLException {
// Відкриваємо БД
String myPath=DB_PATH+DB_NAME;
myDataBase=SQLiteDatabase.openDatabase(myPath,null,SQLiteDatabase.OPEN_READONLY);}

@Override

public synchronized void close() {
if (myDataBase!=null) myDataBase.close();
super.close (); }

@Override

public void onCreate(SQLiteDatabase db) { ... }

@Override

public void onUpgrade(SQLiteDatabase db,int oldVersion,int newVersion) { ...
// Тут можна додати допоміжні методи для доступу і отримання даних з БД
// Тут можна повертати курсори через "return myDataBase.query (....)", це полегшить їх
// використання в створенні адаптерів для ваших view
... }

// Тепер можна створити новий екземпляр класу DataBaseHelper і викликати методи
// createDataBase () і openDataBase (). Не забудьте змінити YOUR_PACKAGE на ім'я
// пакета в вашому додатку в рядку DB_PATH.

DataBaseHelper myDbHelper=new DataBaseHelper (this);
myDbHelper=new DataBaseHelper (this);

try {
myDbHelper.createDataBase (); }
Catch (IOException ioe) {
throw new Error ("Unable to create database"); }

try {
myDbHelper.openDataBase (); }
Catch (SQLException sqle) {
throw sqle; }
.....
```

Спосіб №2. Покладемо файл бази даних в zip-архів і перемістимо його в папку res/raw. У методі onCreate() перевіримо – чи запускається додаток перший раз. Розпакуємо БД і перемістимо її на SD-карту. Встановимо з'єднання з базою. Перевіримо, чи існує файл БД, якщо не існує, то потрібно це файл розгорнути.

Даний метод помістимо в onCreate().

```
public void createDataBase() throws IOException {
// Отримуємо шлях до SD-карти.
File DB_PATH=myContext.getExternalCacheDir();
// Створюємо каталог для нашої бази даних
DB_PATH.mkdirs ();
// Перевіряємо чи є вже файл БД на карті
File db=new File (DB_PATH, DB_NAME);
if (!db.exists()) {
// Якщо файлу немає, то спробуємо його створити
db.createNewFile(); }
try { copyFromZipFile(); }
```

```
Catch (IOException e) { throw new Error("Error copying database",e); }}
```

Цей метод копіює файл БД з res/raw на SD-карту

```
private void copyFromZipFile() throws IOException {  
    InputStream is=myContext.getResources().OpenRawResource(R.raw.database);  
    File outFile=new File(DB_PATH,DB_NAME);  
    OutputStream myOutput=new FileOutputStream(outFile.getAbsolutePath());  
    ZipInputStream zis=new ZipInputStream(new BufferedInputStream(is));  
    try {  
        ZipEntry ze;  
        while ((ze=zis.getNextEntry())!=null) {  
            ByteArrayOutputStream baos=new ByteArrayOutputStream();  
            byte[] buffer=new byte[1024];  
            int count;  
            while ((count=zis.read (buffer))!=-1) { baos.write(buffer,0,count); }  
            baos.writeTo(myOutput); }  
    } finally {  
        zis.close();  
        myOutput.flush();  
        myOutput.close();  
        is.close(); }  
}
```

Встановлюємо з'єднання з БД.

```
public SQLiteDatabase openDataBase() throws SQLException {  
    File DB_PATH=myContext.getExternalCacheDir();  
    File dbFile=new File(DB_PATH,DB_NAME);  
    myDataBase=SQLiteDatabase.openDatabase(dbFile.getAbsolutePath(),null,  
                                           SQLiteDatabase.OPEN_READWRITE);  
    return myDataBase; }
```

Висновки. Після проведеного аналізу існуючих систем керування базами даних було обрано для використання систему SQLite. У ній спрощено підключення баз даних до Android-додатків. Програма є безкоштовною для користувачів і має відкритий код, та можливість роботи з БД, що створені за допомогою інших СКБД. Також за допомогою даного середовища можна принаймні 2-ма способами робити перенесення БД в Android-додатки. Перший спосіб полягає у звичайному копіюванні усієї бази даних у сам додаток, інший же полягає у створенні і записуванні на флеш пам'ять, яка є практично у всіх сучасних мобільних пристроях всієї бази даних а потім через вказування шляху у Android-додатку відбувається підключення до неї.

1. Дейтел П. Android для программистов: создаём приложения / П. Дейтел, Х. Дейтел, Э. Дейтел, М. Моргано. – СПб. : Питер, 2013. – 560 с.: ил.
2. Порменюк М.С. Порівняння засобів перевірки обмежень у реляційних СУБД/ М.С. Порменюк, О.Ю. Безносик // Комп'ютерно-інформаційні технології в багаторівневій вищій освіті. – К., 2014. – С. 282-287.
3. Неполное Руководство по SQLite для пользователей Windows / Перевод: А.Г Пискунов. – 9 октября 2015 г. – 118 с.
4. Офіційний сайт розробників Android [Електронний ресурс]. – <https://developer.android.com>.
5. Пособие пользователя. Работа в среде SQLite [Електронний ресурс]. – http://ukrhosting.ua/posibnik_koristuvacha-p-1.html.

УДК 004.056.55

Красиленко В.Г., к.т.н., с.н.с., доцент, професор; Нікітович Д.В., науковий співробітник
Вінницький інститут Університету "Україна"

МОДЕЛЮВАННЯ БАГАТОКРОКОВИХ ТА БАГАТОСТУПЕНЕВИХ ПРОТОКОЛІВ УЗГОДЖЕННЯ СЕКРЕТНИХ МАТРИЧНИХ КЛЮЧІВ.

Красиленко В.Г., Нікітович Д.В. Моделювання багатокрокових та багаступеневих протоколів узгодження секретних матричних ключів. Розглядаються протоколи узгодження секретного матричного ключа для криптографічних перетворень в системах і моделях матричного типу. Основою таких протоколів є узагальнення відомих протоколів Діффі-Хеллмана на матричний випадок і відповідні матричні процедури для формування двовимірних ключів. Обґрунтовані необхідність та переваги створення, узгодження та застосування матричних ключів для покращених криптографічних систем матричного типу і процедур зашифрування-розшифрування зображень. Запропоновані багатокрокові та багаступеневі протоколи узгодження ключа з метою їх вдосконалення та підвищення стійкості до атак. Для підтвердження достовірності запропонованих удосконалень виконано ряд модельних експериментів у середовищі Mathcad Professional. Показані переваги багатокрокових протоколів узгодження ключа за рахунок використання в них багаступеневих поелементно-матричних піднесень у степінь за модулем та конвеєрних процедур. Моделі та процедури враховують специфіку зображень і легко адаптуються до паралельних реалізацій та новітніх апаратних матричних процесорів. Наведено результати моделювання процесів створення секретних матричних ключів у вигляді зображень великої розмірності (320×240) на основі запропонованих протоколів.

Ключові слова: криптографічні перетворення зображень, матричний алгоритм Діффі-Хеллмана, узагальнені матричні моделі, секретний матричний ключ, багатокроковий протокол, розшифрування, протокол узгодження секретного спільного ключа, піднесення в степінь по модулю.

Красиленко В.Г., Нікітович Д.В. Моделирование многошаговых и многоступенчатых протоколов согласования секретных матричных ключей. Статья посвящена усовершенствованию и моделированию протоколов согласования секретного матричного ключа для криптографических преобразований в системах и моделях матричного типа. Основой таких протоколов является обобщение известных протоколов Диффи-Хеллмана на матричный случай и соответствующие математические процедуры для формирования двумерных ключей. Обоснована необходимость и преимущества создания, согласования и применения матричных ключей для улучшенных криптографических систем матричного типа и процедур зашифрования-расшифровки изображений. Предложены многоступенчатые и многошаговые матричные согласовательные протоколы с целью совершенствования их устойчивости к атакам. Для подтверждения достоверности предложенных протоколов выполнен ряд модельных экспериментов в программной среде Mathcad Professional. Показаны преимущества многоступенчатых поэлементно-матричных вознесений в степень по модулю и использования многошаговых конвейерных процедур. Вычислительные процедуры и матричные модели учитывают специфику изображений и легко адаптируются к параллельным реализациям и новейшим аппаратным матричным процессорам. Приведены результаты моделирования процессов создания секретных матричных ключей в виде изображений большой размерности (320×240) на основе предложенных модификаций протоколов.

Ключевые слова: криптографические преобразования изображений, матричный алгоритм Диффи-Хеллмана, обобщенные матричные модели, секретный матричный ключ, расшифровка, протокол согласования секретного совместного ключа, вознесение в степень по модулю.

Krasilenko V.G., Nikitovich D.V. Simulation of cryptographic transformations of color images based on matrix models of permutations with spectral and bit-plane decompositions. The article is devoted to the improvement and simulation of secret matrix key negotiation protocols for cryptographic transformations in matrix type systems and models. The basis of such protocols is the generalization of the known Diffie-Hellman and others protocols to the matrix case and the corresponding mathematical procedures for the formation of two-dimensional keys. The necessity and advantages of creation, matching and application of matrix keys for improved matrix-type cryptographic systems and image encryption-decryption procedures are substantiated. Multistage and multi step matrix reconciliation protocols are proposed with the aim of improving their resistance to attacks. To confirm the reliability of the proposed protocols a number of model experiments were performed in the software environment of Mathcad Professional. The advantages of multistage and multi step matrix reconciliation protocols are shown. Computational procedures and matrix models take into account the specificity of images and easily adapt to parallel implementations and the latest hardware matrix processors. The results of simulation the creation of secret matrix keys in the form of high-dimensional (320×240) images are presented on the basis of the proposed protocol modifications.

Keywords: Cryptographic image transformations, Diffie-Hellman matrix algorithm, generalized matrix models, secret matrix key, decryption, secret shared key negotiation protocol, modular exponentiation.

Вступ. Широке застосування інформаційних технологій при збільшенні обсягів та значимості інформаційних потоків в епоху електронних комунікацій потребує надійного та ефективного захисту цілісності інформаційних об'єктів (ІО) та стійкості різноманітних захищених систем управління до потенційних загроз. Зросла доля специфічних текстово-графічних документів (ТГД) у вигляді табличних даних, малюнків, діаграм, підписів, віз, резолюцій, тощо, які є зображеннями і

які необхідно опрацьовувати та передавати каналами з обмеженим чи закритим доступом, засвідчувати їх цифровими підписами. Особливе місце серед відомих методів захисту ІО від несанкціонованого доступу займають криптографічні методи, що спираються на властивості самих ІО, а не носіїв ІО та пристроїв їх обробки та передачі, але якими б надійними не були б криптографічні системи, одним з ключових питань їх застосування на практиці є адміністрування ключами, включаючи процеси генерування ключів та їх узгодження електронним шляхом. Від надійності та стійкості до атак процесів створення спільних для обох сторін безпечних ключів залежить рівень безпеки. Перед обміном інформацією користувачам необхідно домовитись чи сформувати спільний безпечний ключ, з якого за необхідності створюються сесійний та низка похідних під-ключів. Відомі протоколи та алгоритми створення ключа при використанні навіть незахищених каналів зв'язку, наприклад алгоритми Діффі-Хелмана, МТІ, STS та інші [1, 2]. Але більшість з них, як і більшість методів та засобів криптографічних перетворень (КП) ІО, зорієнтовані на системи з послідовною криптографічною обробкою сукупності виділених інформаційних блоків за допомогою одного ключа або його під-ключів, що є суто скалярами, незважаючи на їх довжину. Навіть для найкращих симетричних алгоритмів (на основі діючого стандарту AES, IDEA, тощо) довжини блоків та ключів не перевищують 256 бітів, за винятком хіба-що FEAL, RC6 та інших нових, де ці довжини можуть обмежуватись 1К-2К бітами [1]. Проте темпи розвитку методів криптоаналізу та комп'ютерних засобів спонукають до збільшення довжин ключів, вимірності скінченних полів і діапазонів чисел (модулів), які необхідно опрацьовувати, а це позначається на обчислювальній продуктивності, що в свою чергу призводить до пошуку нових операцій та методів їх прискорення, ускладнює процедури визначення, верифікації базових параметрів криптографічних систем (КС), генерування та зберігання ключів. Поява паралельних алгоритмів, а особливо матричних процесорів [2, 3], потребує переорієнтації КП на ці засоби, та пошук і створення моделей матричного типу (МТ), засобів виконання КП ІО у вигляді ТГД (зображень), що краще відображаються на матричні процесори [4-11]. В останні роки збільшується доля задач, для яких КП необхідно виконувати над багатовимірними сигналами, багато-спектральними зображеннями різних фізичних, аерокосмічних об'єктів, а це потребує не лише матричних моделей (ММ) КП, але й однорідних до їх структури секретних матричних, тензорних ключів, наприклад у вигляді матриць-зображень [6-16].

Аналіз останніх досліджень і публікацій. Переваги КП матричними алгоритмами на основі більш узагальнених матричних афінних шифрів, в тому числі при створенні сліпих цифрових підписів на ТГД були продемонстровані в [4-7]. Ще більш узагальнені матричні афінно-перестановочні шифри були запропоновані та досліджені в [8], базовою операцією яких є матричні моделі перестановок (ММ_П), які мають наочну простоту. Зростання долі робіт, що присвячені КП, зашифруванню та розшифруванню різноманітних кольорових багато-спектральних зображень [6, 8-12, 17-20], теж підтверджує актуальність дослідження та удосконалення ММ КП та ставить, як показує аналіз цих робіт, на порядок денний гостру проблему створення для таких ММ і відповідних матричних ключів (МК). Для матрично-афінно-перестановочних алгоритмів [8] необхідно мати два види МК: набір бінарних матриць перестановок, позначимо тут їх як МК_П, що потрібні і для ММ_П [10-12, 17-20], та МК загального типу у вигляді чорно-білого чи кольорового (детермінованого, випадкового чи псевдо-випадкового) зображення, позначимо тут як МК_З (від «зображення»). Частково питання щодо формувань, застосувань МК_П розглядалися в [8-10, 17-19, 21], добре досліджені в теорії груп та лінійній алгебрі, а тому тут ми їх розглядати не будемо, а специфіку їх застосування для протоколів узгодження секретного ключа розглянемо в іншій роботі. Стосовно МК_З відмітимо, що в роботі [14] була запропонована модифікація алгоритму Діффі-Хелмана на матричний випадок для створення 2-D ключа, а в [15] були розглянуті алгоритми формування двовимірних ключів для матричних алгоритмів криптографічних перетворень зображень, виконане їх часткове моделювання та запропонована ідея застосування і у матричному випадку покращених підходів та методів організації прискорених обчислень на основі матричної паралельної логіки. Але в цих роботах не проводились модельні експерименти зі збільшеною розмірністю МК_З та не були показані шляхи усунення відомих недоліків алгоритму Діффі-Хелмана.

Постановка наукової проблеми. Тому метою роботи є подальше вдосконалення протоколів узгодження секретного матричного ключа для ММ КП у системах та моделях матричного типу, покращення алгоритмів формування більш якісних та стійких до атак матричних ключів типу МК_3, їх моделювання та дослідження.

Виклад основного матеріалу та результатів дослідження.

Теоретична частина. Наведений у [15] протокол (базовий!) виконується так. Сторони узгоджують між собою чи й іншими користувачами мережею зв'язку матрицю-основу **K**, наприклад матрицю, що відповідає чорно-білому зображенню вибраного розміру. Бажано попередньо додаванням до неї матриці **R**, всі елементи якої дорівнюють «1» змістити її значення елементів у діапазон 1-256 або, як буде показано нижче, навіть у діапазон 2-256, щоб у результуючій матриці не було «0» та «1» елементів. Перша сторона, наприклад, абонент X (Alisa!), вибирає з множини зображень чи генерує відомим методом, засобом випадкове зображення матрицю **S_AS-R**, шляхом додавання до неї матриці **R** зміщує її значення елементів у діапазон 1-256, отримуючи скореговану матрицю **A** (у деяких експериментах позначена як **S_AS**) та обчислює 2-D масив (матрицю) **KSA** за формулою $KSA \equiv K^A \bmod (m1 \cdot R)$, де матриці **A**, **K**, **R** та **KSA** мають однакові розміри $I \times J$, а операція піднесення у степінь за модулем $m1$ є по-елементною, тобто $KSA_{i,j} \equiv K_{i,j}^{A_{i,j}} \bmod m1$. Цю матрицю шляхом віднімання від неї матриці **R** він коригує у стандартного формату зображення (**KSA-R**) та відправляє його другому абоненту Y (Bob!). Абонент Y взявши інше зображення **S_BS-R** та аналогічним чином коригуючи його у матрицю **B**, обчислює матрицю **KSB** за формулою $KSB \equiv K^B \bmod (m1 \cdot R)$ та відправляє аналогічним чином скориговане відповідне йому зображення (**KSB-R**) першому абоненту X. Перший абонент X, отримавши це зображення коригує його у матрицю **KSB** (у подальшому пряме та обернене коригування матрицею **R** будемо опускати, розуміючи зв'язок між матрицями та зображеннями) та, використовуючи лише йому відому матрицю **A**, обчислює значення матричного ключа $KA \equiv KSB^A \bmod (m1 \cdot R) \equiv K^{B \cdot A} \bmod (m1 \cdot R)$. Аналогічними діями абонент Y обчислює значення матричного ключа $KB \equiv KSA^B \bmod (m1 \cdot R) \equiv K^{A \cdot B} \bmod (m1 \cdot R)$. Таким чином сторони отримують таємний ключ $KEY \equiv K^{A \cdot B} \bmod (m1 \cdot R)$, рівність якого забезпечена протоком для обох сторін. Вони можуть використовувати його як для зашифрування та розшифрування при передачі 2-D даних, зображень, особливо для ММ КП та у КС МТ [6-8,13,16,20], тощо, так і використати у якості матриці-основи для оновлень секретного ключа. Відомо, що не існує жодного ефективного алгоритму розв'язування задачі обчислення дискретного логарифма за модулем, а тому з урахуванням розширення та ускладнення задачі на матричний випадок, робить її розв'язування для зловмисника ще складнішим. Тому наведений протокол є взаємно безпечним. Якщо ж матриця **K** є невідомою для третьої сторони, як ми зазначали вище, то тоді цей протокол унеможлиблює і так звану атаку «людина всередині». В загальному випадку діапазон значень елементів всіх використовуваних матриць у ММ такого протоколу може бути достатньо значним, що призводить до необхідності передавати по каналах зв'язку самі матриці, хоч і зменшує візуальне сприйняття. Вихід і у цих випадках можна знайти, якщо матриці представляти не одним а набором зображень, кожне з яких буде відповідати різним позиціям байтових багато-розрядних представлень значень елементів матриць. Якщо матриця **K** є публічною, то для підсилення стійкості протоколу сторони можуть лише відомим їм, наприклад, більш раннім ключем закривати її одним з вищевказаних шифрів [6-8, 20], що призводить фактично до чергового оновлення ключа. Крім того, можна закривати і матриці **A** та **B**, а в якості останніх чи в якості ключів закриття можуть бути використані характерні лише для кожної сторони зображення, що є по суті їх ідентифікаторами.

Сутність **багатокрокового протоколу** полягає у повторенні процедурних дій простого (одно-крокового) протоколу декілька разів по домовленості сторін. Використовуючи для наступних кроків в якості матриці-основи інші зображення, наприклад, отримані у попередніх кроках ключі чи результуючі степені за модулем, а у якості матриць-степенів обчислені їх попередні аналогічні степені чи деякі вибрані з певного набору зображення, сторони формують низку спільних ключів. Оскільки зловмиснику важче перехопити всі повідомлення, що

пересилаються сторонами та можуть містити закриті дані стосовно вибраних часу та інших ідентифікаторів сторін, вгадати час сеансів обміну, та необхідно розв'язати значно більшу кількість задач обчислення дискретного логарифма за модулем фактично для кожного елемента відповідних всіх 2-D масивів, то це підвищує стійкість багатокрокового протоколу у порівнянні з простим. Цьому сприяє і застосування конвеєрних багаторазових піднесення у степінь за модулем, і гнучкість вибору сторонами чергових матриць для процедур зі збільшеної по потужності їх множини. Розглянемо **теоретичні основи** та ММ одного з таких запропонованих варіантів удосконалення протоколу. Позначимо отриманий після першого кроку сторонами ключ як **KeyA1 (KeyB1)**, де букви відповідають стороні, а цифра – кроку. Тоді, взявши його як матрицю основу та виконавши необхідні аналогічні вищеописаним процедури протоколу, сторони після другого кроку отримають ключі: **KeyA2** $\equiv$ **KeyA1**^{B*A} mod (m1·R) (**KeyB2** $\equiv$ **KeyB1**^{B*A} mod (m1·R)), а оскільки ключі-основи **KeyA1** та **KeyB1** рівні, то рівними є і ключі **KeyA2**, **KeyB2**, тобто **KeyA2** $\equiv$ **KeyB2**. А далі аналогічно: після 3-го кроку будуть обчислені ключі **KeyA3** $\equiv$ **KeyA2**^{B*A} mod (m1·R) (**KeyB3** $\equiv$ **KeyB2**^{B*A} mod (m1·R)) і т. д., при цьому буде забезпечена рівність покровових ключів. Провівши деякі нескладні математичні перетворення, які тут не показані, можна встановити, що після виконання q кроків отримані ключі будуть виражені наступними виразами для обох сторін: **KeyAq** $\equiv$ **K**^{B^q*A^q} mod (m1·R) та **KeyBq** $\equiv$ **K**^{B^q*A^q} mod (m1·R), де кожен елемент відповідних матриць **A**^q **B**^q є q- а степінь (не за модулем!) кожного відповідного елемента матриць **A** та **B**. З цих виразів можна зробити цікаве припущення, яке потребує підтвердження і модельними експериментами, що замість взаємних передач проміжних повідомлень на кожному кроці можна виконати відповідні піднесення матриці **KSA** $\equiv$ **K**^A стороною A у q- у степінь, а матриці **KSB** $\equiv$ **K**^B стороною B у q- у степінь, а потім після обміну ними та піднесення їх сторонами у свої матриці степені **A** та **B** так само повторити піднесення останніх у q- у степінь. Іншими словами процедури піднесення у степінь можуть бути ієрархічними багатоступеневими, в тому числі і за модулем чи матрицею модулів, а багатоступеневість тісно пов'язана з повторенням кроків, тобто з багатокроковим аспектом запропонованих модифікацій протоколу.

Розглянемо деякі аспекти, які необхідно враховувати при використанні такого протоколу. По-перше, оскільки в якості основи можуть бути використані різні з відомої множини матриць чи з отриманої множини попередніх ключів, то виникає потреба в уточненні сторонами при колізіях цієї основи. Один з можливих варіантів полягає у застосуванні надійного та відомого для сторін ключа для шифрування ним та перевірки основи. Якщо до зображення цього надійного ключа інша сторона створить комплементарне зображення AD-R, так щоб їх сума дорівнювала m1·R, то на основі малої теореми Ферма, після за шифрування стороною X матриці основи **K** піднесенням її у матрицю степінь **S_AS** та передачі результату **KSA-R** сторона Y підносить матрицю **K** у степінь **S_BS=AD**, отриману від сторони X матрицю **KSA** множить по-елементно на обчислену **KSB** за модулем і отримує матрицю **F**, що повинна співпадати з основою **K**, дивись відповідні зображення на рис.1, отримані у результаті першого базового експерименту. Якщо злоумисник не знав **K**, то перехопивши **KSA** він ніяк не зможе знайти **S_AS** чи по аналогії з **KSB** знайти **S_BS**. Якщо ж він знає **K**, то згідно теорії розв'язування задачі обчислення дискретного логарифма за модулем є складним процесом, проте, як показав наш експеримент (рис.1), при невеликих значеннях чисел у матрицях (у наших зображеннях) у деяких окремих випадках, хоч і не повністю, вдається відновити зображення **S_AS**, зламавши брутальною атакою. По-друге, враховуючи перше та необхідність покращення стійкості протоколу до можливих атак, особливо для випадків, коли основа **K** є відомою, бажано ускладнити задачу злоумиснику. А це можна зробити, застосувавши вищерозглянуті ієрархічні піднесення у степінь за модулем, тим більше, що вони пов'язані з багатокроковими протокольними процедурами. Крім того, навіть і матриці **A** та **B** і всі проміжні матриці можна підносити за додатковими простими домовленостями у якісь скалярні чи навіть матричні степені, що дуже сильно за рахунок вкладених ієрархій ускладнює задачу злоумиснику.

Експериментальна частина. На рис. 1-9 зображені результати моделювання запропонованих багатокрокових багатоступеневих протоколів узгодження секретного ключа. Зображення, що були використані для моделювання варіантів реалізації таких протоколів та для сприятливого відображення всіх проміжних дій та кроків показані на рис. 2, там же частково і результатні ключі: KA_4-R, тощо. На рис.3 показані програмні модулі-формули, що використовувались, як один з можливих без прискорень варіант піднесення матриць у матричні степені за матрицею модулів та для імітації брутальної атаки злоумисником для обчислення степені з перехопленого повідомлення. Як видно з рис.4. для більш якісного, навіть на візуальному

рівні закриття зображень, їх перетворення різними багатоступеневими піднесеннями у степені за модулями кращим є перший варіант, а для майже всіх варіантів бажано, щоб у матрицях були відсутні елементи зі значеннями «0» та «1», оскільки вони при перетвореннях ну будуть змінюватись. Особливу увагу треба приділяти «0», оскільки після коригувань вони стають «1». Експерименти показали, що краще генерувати відомими у пакеті інструментами псевдо-випадкові зображення зразу у відповідності до встановлених меж діапазону значень, проте для відображення результатів і їх кращого аналізу тут ми наводимо саме такі вибрані нами для демонстрації правильних проміжних, кінцевих результатів.

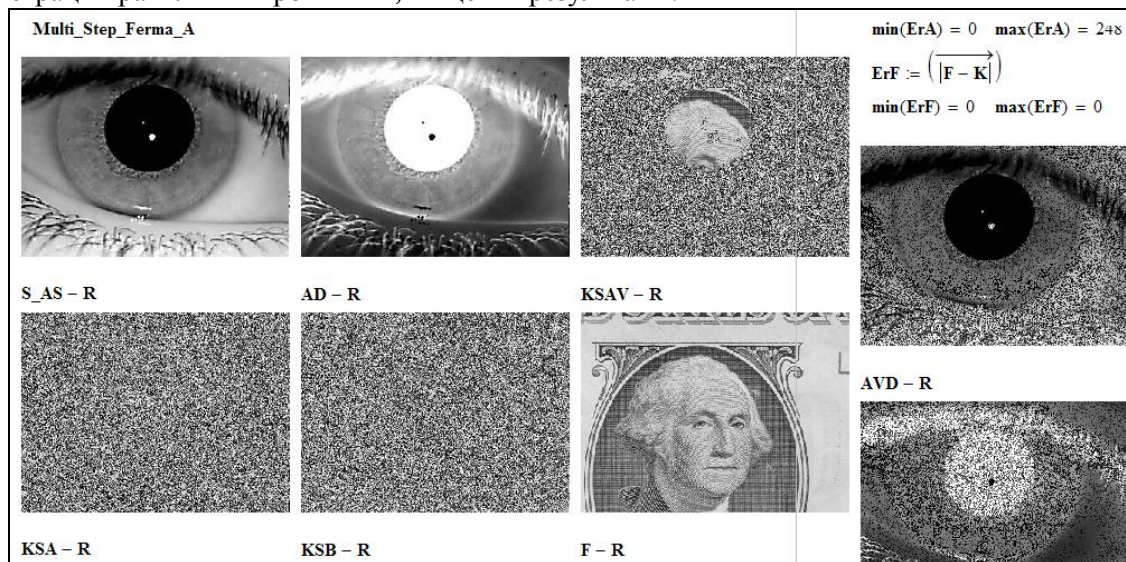


Рис.1. Вигляд вибраних для моделювання зображень S_AS-R з комплементарним до нього (негатив!) AD-R, зображень утворених шляхом матричного піднесення зображення-основи K-R у відповідні матриці-степені (S_AS-R та AD-R) матриць-зображень KSA-R та KSB-R, верифікаційного зображення F-R та розрахованого (на основі K і KSA-R) імітацією брутальної атаки зображення AVD-R. Фрагменти вікна Mathcad (базовий модельний експеримент).

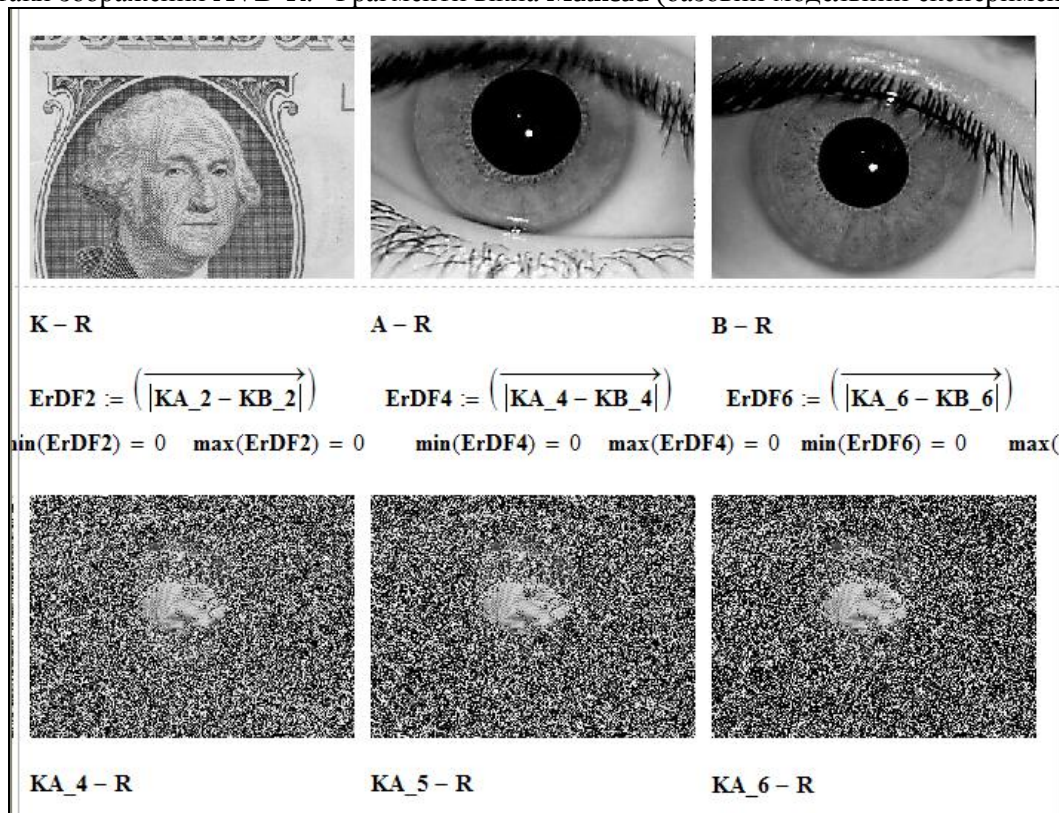


Рис. 2. Фрагмент з Mathcad. Вигляд вибраних для моделювання багатокрокового протоколу зображень: зображення-основи K-R, довільних ідентифікаторів-зображень A-R та B-R відповідно сторони A та сторони B, утворених протоколом повідомлень та ключів (нижній ряд) у вигляді зображень та формули для перевірки рівності ключів після кожного кроку (1,2,3-го).
На рис.5 показані програмні модулі-формули з вікна Mathcad, що використовувались при моделюванні багатокрокового узагальненого протоколу узгодження секретного матричного ключа сторонами A (ліворуч) та B (праворуч). Показані перший (1 та 2 дії обох сторін, згори вниз) та другий (3 та 4 дії обох сторін, згори вниз) кроки. Отриманими масивами KA та KB з непарними номерами сторони обмінюються, а відповідні масиви з парними номерами використовуються як основа для подальших кроків або як отриманий (отримані!) ключі. Вигляд утворених та отриманих сторонами зображень показані на рис.6. Вони свідчать про зміну ключів при переходах та різницю між переданими сторонами повідомленнями після непарних дій та рівності покровкових ключів (темні зображення!).

$\begin{aligned} \text{KSA}_{i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow K_{i,j} \\ &\text{while } 1 < S_AS_{i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot K_{i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$	$\begin{aligned} \text{KSB}_{i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow K_{i,j} \\ &\text{while } 1 < S_BS_{i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot K_{i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$
a)	б)
$\begin{aligned} \text{AVD}_{i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow K_{i,j} \\ &\text{while } 1 < 256 \wedge s \neq \text{KSA}_{i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot K_{i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &1 \end{aligned}$	$F_{i,j} := \text{mod}(\text{KSA}_{i,j} \cdot \text{KSB}_{i,j}, m1)$
в)	г)

Рис. 3. Програмні модулі-формули з вікна Mathcad, що використовувались при моделюванні операцій поелементно-матричного піднесення у степінь за модулем (матриця K-основа, S_AS та S_BS – матриці-степені, m1- модуль), множення за модулем матриць та знаходження матриці-степені AVD на основі K та перехопленого повідомлення KSA.

Multi_Step_Ferma_A $\begin{aligned} \text{A_St}_{1,i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow K_{i,j} \\ &\text{while } 1 < S_AS_{i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot K_{i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$ $\begin{aligned} \text{A_St}_{2,i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow K_{i,j} \\ &\text{while } 1 < \text{A_St}_{1,i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot K_{i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$ $\begin{aligned} \text{A_St}_{3,i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow K_{i,j} \\ &\text{while } 1 < \text{A_St}_{2,i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot K_{i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$ $\begin{aligned} \text{A_St}_{4,i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow K_{i,j} \\ &\text{while } 1 < \text{A_St}_{3,i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot K_{i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$	$\begin{aligned} \text{B_St}_{1,i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow S_BS_{i,j} \\ &\text{while } 1 < S_BS_{i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot S_BS_{i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$ $\begin{aligned} \text{B_St}_{2,i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow \text{B_St}_{1,i,j} \\ &\text{while } 1 < \text{B_St}_{1,i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot \text{B_St}_{1,i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$ $\begin{aligned} \text{B_St}_{3,i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow \text{B_St}_{2,i,j} \\ &\text{while } 1 < \text{B_St}_{2,i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot \text{B_St}_{2,i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$ $\begin{aligned} \text{B_St}_{4,i,j} &:= 1 \leftarrow 1 \\ &s \leftarrow \text{B_St}_{3,i,j} \\ &\text{while } 1 < \text{B_St}_{3,i,j} \\ &\quad \left \begin{aligned} &s \leftarrow \text{mod}(s \cdot \text{B_St}_{3,i,j}, m1) \\ &1 \leftarrow 1 + 1 \end{aligned} \right. \\ &s \end{aligned}$
a)	б)

Рис.4. Програмні модулі-формули з вікна Mathcad, що використовувались при моделюванні багатокрокового поелементно-матричного піднесення у степінь за модулем: а) –варіант 1 піднесення матриці основи K у послідовно створені нові матричні степені за модулем (ліворуч!) та

варіант 2 піднесення степеня $B_St_1(2,3,\dots)$ матриці S_BS у послідовно створені нові матричні степені за модулем (праворуч!); б)- зображення, отримані після таких піднесень (ліворуч для 1-го варіанту та праворуч для 2-го варіанту).

Отримані результати моделювання процесу узгодження секретного ключа багатокроковим протоколом свідчать, про коректність моделей та правильну роботу проколу та його модифікацій на різних кроках перетворень. Результати моделювання, що наведені на рис.7, та їх порівняння з результатами на рис.6 показують наскільки кращими є отримані ключі у випадку усунення небажаних темних («0» та «1») значень з матриць-зображень. Вони також засвідчують правильність висловлених вище припущень та відповідність теоретичним положенням і правильну роботу протоколу.

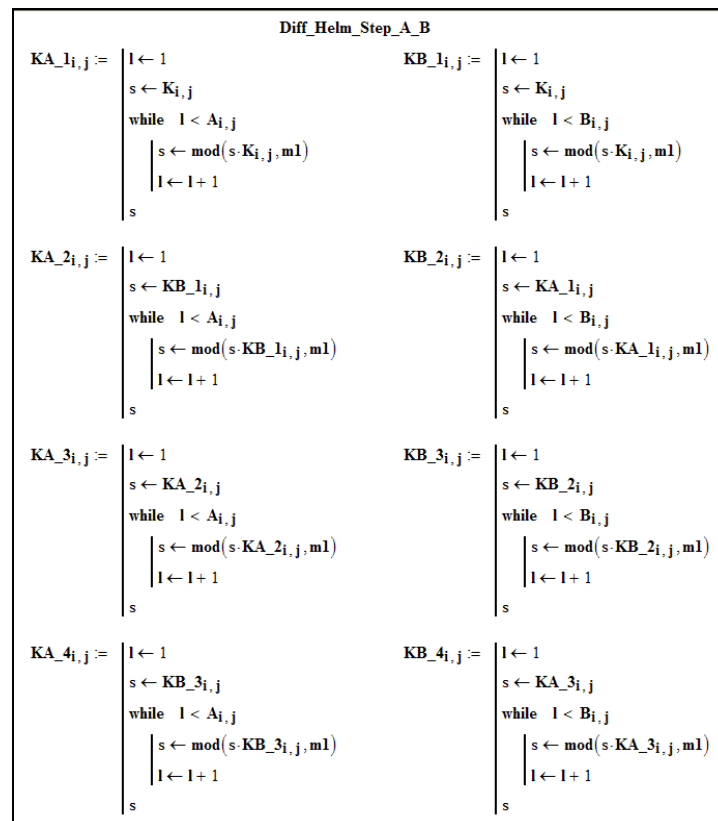


Рис.5. Програмні модулі-формули з вікна Mathcad, що використовувались при моделюванні багатокрокового узагальненого протоколу узгодження секретного матричного ключа сторонами А (ліворуч) та В (праворуч) . Показані перший (1 та 2 дії обох сторін, згори вниз) та другий (3 та 4 дії обох сторін, згори вниз) кроки. Отриманими масивами КА та КВ з непарними номерами сторони обмінюються, а відповідні масиви з парними номерами використовуються як основа для подальших кроків або як отриманий (отримані!) ключі.

Програмні модулі-формули з вікна Mathcad, що використовувались при моделюванні для верифікації сторонами протокольного обміну проміжних повідомлень та створених матричних ключів на попередніх кроках багатокрокового протоколу узгодження секретного ключа для варіанту застосування попередніх, відомих сторонам, проміжних прийнятих повідомлень та ключів, показані на рис.8, а відповідні зображення-результати отримані за їх допомогою показані на рис.9. Розробленою нами раніше та наведеною у наших попередніх роботах програмою-модулем було проведено вимірювання ентропії секретних матричних ключів та проміжних матриць-зображень з метою встановлення суттєвості змін статистичних характеристик і гістограм них особливостей. Експериментами встановлено, що ентропія K була рівна 7,194, для А дорівнювала 7,373, для В – 7,128, а для всіх варіантів та видів степеневих перетворень ентропії проміжних та результуючих матриць-зображень мало змінювалась на всіх кроках та діях та знаходились завжди у межах від 7,73 до 7,83. Зауважимо, що протокол ніяким чином не змінюється при зміні розмірів матриць чи зображень, може бути повністю паралельним, і має підвищену стійкість за рахунок застосування ієрархічних багато-ступеневих піднесень тим паче у матричні і не лише однорідні а й неоднорідні масиви модулів. Довжина отриманих тут у експериментах ключів при переводі їх у скалярний вимір становить $320 \cdot 240 = 76800$ байтів !

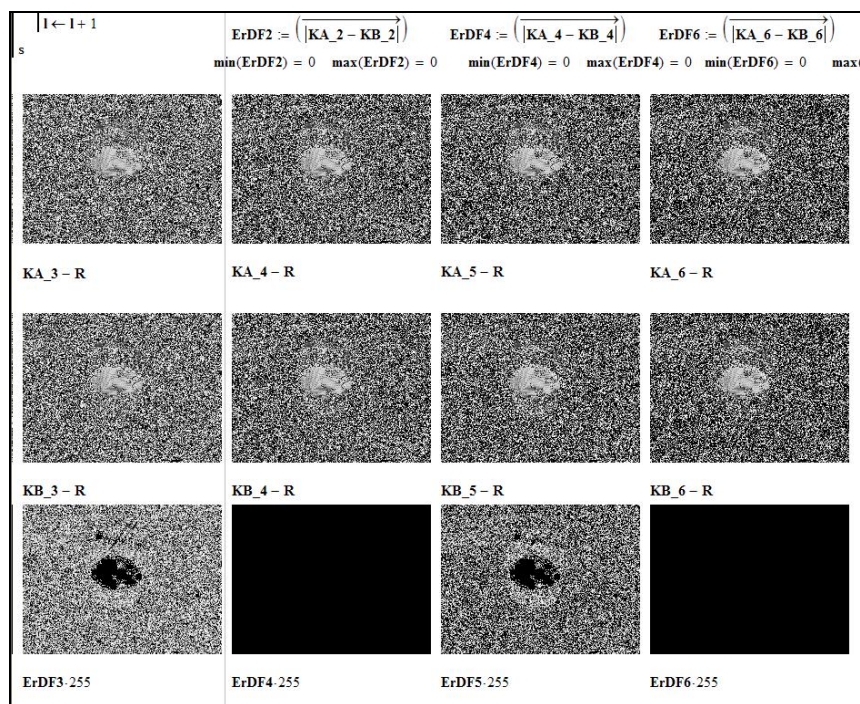


Рис.6. Вигляд утворених та отриманих сторонами зображень (переданих повідомлень KA_3M_R та KB_3M_R, KA_5M_R та KB_5M_R, відповідно після 3-ої, 5-ої дій), секретного ключа (KA_4M_R = KB_4M_R) на другому кроці (3 та 4 дії), ключа (KA_6M_R = KB_6M_R) на третьому кроці (після 5 та 6 дії) (перший та наступні після 3-ого кроки не показані!) та різницевих перевірочних зображень (нижній ряд), що свідчать про їх зміну при переходах та різницю між переданими сторонами повідомленнями після непарних дій та рівності покровових ключів (темні зображення!). Фрагменти вікна Mathcad (модельний експеримент 1).

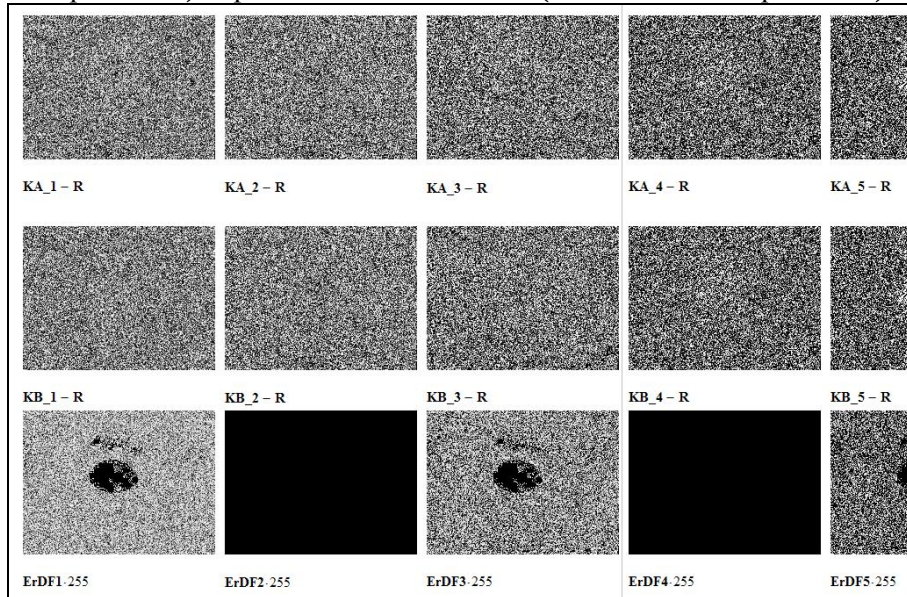


Рис.7. Вигляд утворених та отриманих сторонами зображень (переданих повідомлень KA_1M_R та KB_1M_R, KA_3M_R та KB_3M_R, KA_5M_R та KB_5M_R відповідно після 1-ої, 3-ої, 5-ої дій), секретного ключа (KA_2M_R = KB_2M_R) на першому кроці (1 та 2 дії), ключа (KA_4M_R = KB_4M_R) на другому кроці (після 3-го та наступних кроків не показані!) та різницевих перевірочних зображень (нижній ряд), що свідчать про їх зміну при переходах та різницю між переданими сторонами повідомленнями після непарних дій та рівності покровових ключів (темні зображення!). Фрагменти вікна Mathcad (модельний експеримент 2).

$\begin{aligned} &KA_5M_{i,j} := \\ & \quad l \leftarrow 1 \\ & \quad s \leftarrow KA_4i,j \\ & \quad \text{while } l < KA_3i,j \\ & \quad \quad \left \begin{array}{l} s \leftarrow \text{mod}(s \cdot KA_4i,j, m1) \\ l \leftarrow l + 1 \end{array} \right. \\ & \quad s \end{aligned}$	$\begin{aligned} &KB_5M_{i,j} := \\ & \quad l \leftarrow 1 \\ & \quad s \leftarrow KB_4i,j \\ & \quad \text{while } l < KA_3i,j \\ & \quad \quad \left \begin{array}{l} s \leftarrow \text{mod}(s \cdot KB_4i,j, m1) \\ l \leftarrow l + 1 \end{array} \right. \\ & \quad s \end{aligned}$
$\begin{aligned} &KA_6M_{i,j} := \\ & \quad l \leftarrow 1 \\ & \quad s \leftarrow KB_5M_{i,j} \\ & \quad \text{while } l < KB_3i,j \\ & \quad \quad \left \begin{array}{l} s \leftarrow \text{mod}(s \cdot KB_5M_{i,j}, m1) \\ l \leftarrow l + 1 \end{array} \right. \\ & \quad s \end{aligned}$	$\begin{aligned} &KB_6M_{i,j} := \\ & \quad l \leftarrow 1 \\ & \quad s \leftarrow KA_5M_{i,j} \\ & \quad \text{while } l < KB_3i,j \\ & \quad \quad \left \begin{array}{l} s \leftarrow \text{mod}(s \cdot KA_5M_{i,j}, m1) \\ l \leftarrow l + 1 \end{array} \right. \\ & \quad s \end{aligned}$

Рис.8. Програмні модулі-формули з вікна Mathcad, що використовувались при моделюванні для верифікації сторонами протокольного обміну проміжних повідомлень та створених матричних ключів на попередніх кроках багатокрокового протоколу узгодження секретного ключа. Показано режим верифікації після першої та другої дії на третьому кроці.

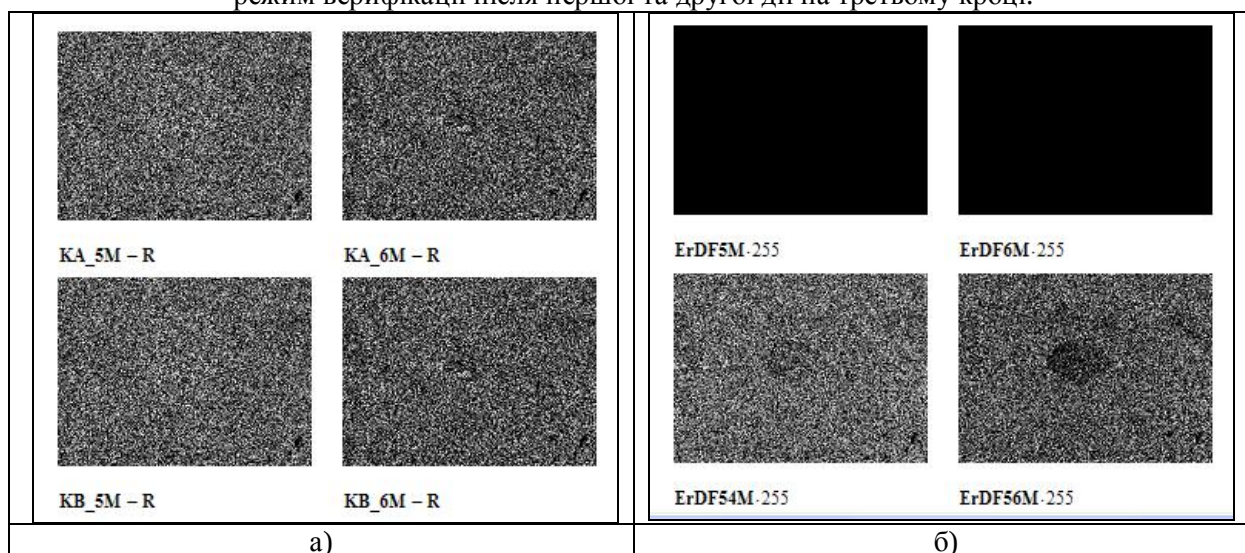


Рис.9. Вигляд утворених та отриманих сторонами зображень (верифікаційних KA_5M_R та KB_5M_R повідомлень), секретного ключа (KA_6M_R = KB_6M_R) на третьому кроці – а) та різницевих перевірочних зображень –б), що свідчать про їх зміну при переходах від 4-ої до 5-ої дії , від 5-ої до 6-ої дії (внизу праворуч) та про рівність зображень, утворених сторонами після 5-ої та 6-ої дій (вгорі праворуч).

Висновки Обґрунтовані необхідність та переваги створення, узгодження та застосування матричних ключів для покращених криптографічних систем матричного типу і процедур зашифрування-розшифрування зображень. Запропоновані багатокрокові та багатоступеневі протоколи узгодження ключа з метою їх вдосконалення та підвищення стійкості до атак. Для підтвердження достовірності запропонованих удосконалень виконано ряд модельних експериментів у середовищі Mathcad Professional. Показані переваги багатокрокових протоколів узгодження ключа за рахунок використання в них багатоступеневих поелементно-матричних піднесень у степінь за модулем та конвеєрних процедур. Моделі та процедури враховують специфіку зображень і легко адаптуються до паралельних реалізацій та новітніх апаратних матричних процесорів. Наведено результати моделювання процесів створення секретних матричних ключів у вигляді зображень великої розмірності (320 * 240) на основі запропонованих протоколів. Довжина отриманих тут у експериментах ключів при переводі їх у скалярний вимір становить 320*240=76800 байтів !

2. Коркішко Т. Алгоритми та процесори симетричного блокового шифрування / Т. Коркішко, А. Мельник, В. Мельник. – Львів: БаК, 2003. – 168 с.
3. Красиленко В.Г. Алгоритми та архітектури для високоточних матрично-матричних перемножувачів на основі оптичної чотирьохзначної знакозмінної арифметики / В.Г. Красиленко // Вимірювальна та обчислювальна техніка в технологічних процесах. - 2004. - №1.- С. 13-26.
4. Красиленко В.Г. Розробка методу криптографічного захисту інформації тексто-графічного типу / В.Г. Красиленко, С.А. Свіренюк // Наука і навчальний процес: науково-методичний збірник НПК. – Вінниця, 2006. – С. 73-74.
5. Красиленко В.Г. Моделювання матричних алгоритмів криптографічного захисту / В.Г. Красиленко, Ю.А. Флавицька // Вісник НУ «Львівська політехніка» «Комп'ютерні системи та мережі». - № 658. – С. 59-63.
6. Красиленко В.Г. Моделювання матричних афінних алгоритмів для шифрування кольорових зображень / В.Г. Красиленко, К. Огородник, Ю. Флавицька // Комп'ютерні технології: наука і освіта. Тези доповідей V Всеукр. наук.-пр. конф. – Київ, 2010. – С. 120-124.
7. Красиленко В.Г., Матричні афінні шифри для створення цифрових сліпих підписів на текстографічні документи / В.Г. Красиленко, С.К. Грабовляк // Системи обробки інформації. – Х.: ХУПС, 2011. – Вип. 7(97). – С. 60 – 63.
8. Красиленко В.Г. Матричні афінно-перестановочні шифри для шифрування та дешифрування зображень / В.Г. Красиленко, С.К. Грабовляк // Системи обробки інформації. - Х.: ХУПС, 2012. – Вип. 3 (101).-т. 2. – С. 53-62.
9. Красиленко В.Г. Матричні моделі криптографічних перетворень зображень з матрично-бітово-зрізовою декомпозицією і перемішуванням та їх моделювання / В. Г. Красиленко, Д.В. Нікітович //Матеріали 68 НТК «Сучасні інформаційні системи і технології. Інформаційна безпека», ч. 3, секції 3-4. – Одеса, ОНАЗ ім. О.С.Попова, 2013. – С.139-143.
10. Красиленко В.Г. Криптографічні перетворення зображень на основі матричних моделей перестановок з матрично-бітово-зрізовою декомпозицією та їх моделювання / В. Г. Красиленко, В. М. Дубчак // Вісник Хмельницького національного університету. Технічні науки. - 2014. - № 1. - С. 74-79.
11. Красиленко В.Г. Моделювання модифікованих матричних моделей криптографічних перетворень зображень з верифікацією цілісності криптограм / В.Г. Красиленко, Д.В. Нікітович // Матеріали II міжнародної науково-практичної Інтернет-конференції «Інформаційні технології: теорія, інновації, практика». – Полтава: ПолтНТУ, 2015. – С. 86-89.
12. Красиленко В.Г. Моделювання криптографічних перетворень зображень на основі їх матрично-бітово-зрізової декомпозиції та матричних моделей перестановок з верифікацією цілісності / В.Г. Красиленко, Д.В. Нікітович // Всеукраїнська науково-практична конференція молодих вчених та студентів «Перспективні напрямки сучасної електроніки, інформаційних і комп'ютерних систем» (MEICS-2015). – Дніпропетровськ: Дніпропетровський національний університет ім. Олеся Гончара, 2015. - С. 32-34.
13. Красиленко, В.Г. Моделювання матричних афінних алгоритмів для шифрування кольорових зображень / В. Г. Красиленко, К. В. Огородник, Ю.А.Флавицька // Комп'ютерні технології: наука і освіта: тези доповідей V Всеукр. наук.-пр. конф. – К., 2010. – С.120-124.
14. Красиленко В.Г. Моделювання модифікованого алгоритму створення 2-D ключа в криптографічних застосуваннях / В.Г. Красиленко, О.І. Нікольський, О.О. Лазарєв // Науково-методичний збірник НПК «Наука і навчальний процес». – Вінниця, 2008. – С. 107-109.
15. Красиленко В. Г. Алгоритми формування двовимірних ключів для матричних алгоритмів криптографічних перетворень зображень та їх моделювання / В. Г. Красиленко, В. І. Яцковський, Р. О. Яцковська // Системи обробки інформації. - 2012. - Вип. 8. - С. 107-110.
16. Красиленко В. Г. Модифікації системи RSA для створення на її основі матричних моделей та алгоритмів для зашифрування та розшифрування зображень / В. Г. Красиленко, С. К. Грабовляк // Системи обробки інформації. - Х.: ХУПС, 2012. - Вип. 8. - С. 102-106.
17. Красиленко В.Г. Моделювання та дослідження криптографічних перетворень зображень на основі їхньої матрично-бітово-зрізової декомпозиції та матричних моделей перестановок з верифікацією цілісності / В.Г. Красиленко, Д.В. Нікітович // Електроніка та інформаційні технології: збірник наукових праць. – Львів: Львівський національний університет імені Івана Франка, 2015. – Вип. 6. – С 111-127. – Режим доступу: http://elit.lnu.edu.ua/pdf/6_12.pdf
18. Красиленко В.Г. Моделювання криптографічних перетворень кольорових зображень на основі матричних моделей перестановок зі спектральною та бітово-зрізовою декомпозиціями / В.Г. Красиленко, Д.В. Нікітович // Комп'ютерно-інтегровані технології: освіта, наука, виробництво : наук. журн. – Луцьк: Видавництво Луц. нац. техн. ун-т., - 2016. - № 23. - С. 31-36. – Режим доступу: <http://ki.lutsk-ntu.com.ua/node/132/section/9>
19. Красиленко В.Г. Моделювання криптографічних перетворень кольорових зображень з верифікацією цілісності криптограм на основі матричних моделей перестановок / В.Г. Красиленко, Д.В. Нікітович// Матеріали науково-практичної інтернет-конференції «Проблеми моделювання та розроблення інформаційних систем». – Дрогобич : ДДПУ ім. І. Франка, 2016. – С. 128-136. Режим доступу: http://ddpu.drohobych.net/wp-content/uploads/2016/04/material_konf.pdf37
20. Красиленко В.Г. / В.Г. Красиленко, Д.В. Нікітович Моделювання матричних афінних шифрів для криптографічних перетворень зображень // Інформатика та системні науки (ІСН-2017): матеріали VIII Всеукраїнської науково-практичної конференції за міжнародною участю, (м. Полтава, 16–18 березня 2017 року) / за ред. О.О.Ємця – Полтава: ПУЕТ, 2017. – Режим доступу: <http://dspace.puet.edu.ua/handle/123456789/5558>
21. Krasilenko V.G. A noise-immune cryptographic information protection method for facsimile information transmission and the realization algorithms [Text] / V.G. Krasilenko, V.F. Bardachenko, A.I. Nikolsky, et. al., // Proc.SPIE, 2006. - Vol. 6241. – P. 316-322.

УДК 004.002

Лавренчук С.В., Грідін О.О.

Луцький національний технічний університет

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ДИПЛОМНИКІВ ТА ВИПУСКНИКІВ КАФЕДРИ

Лавренчук С.В., Грідін О.О. Інформаційна система для дипломників та випускників кафедри. Описано створення адаптивного сайту для кафедри комп'ютерної інженерії, у якому реалізований повноцінний функціонал та можливість використання на мобільних пристроях. З великою кількістю переваг сайт дає можливість зручного управління та моніторингу за контентом, користувачами, окремими записами при використанні розробленої адміністративної панелі. На базі реалізації сайту досягнуто поставлених цілей: можливості реєстрації та авторизації у системі, надання необхідної інформації студентам-дипломникам, своєчасний контроль та дотримання графіку виконання дипломного проекту, швидкий та зручний спосіб пошуку потрібних матеріалів.

Ключові слова: PHP, контент, сайт, адмін панель, база даних, інформаційна система.

Лавренчук С.В., Грідін О.О. Информационная система для дипломников и выпускников кафедры. Описано создание адаптивного сайта для кафедры компьютерной инженерии, в котором реализован полноценный функционал и возможность использования на мобильных устройствах. С большим количеством преимуществ сайт дает возможность удобного управления и мониторинга за контентом, пользователями, отдельными записями при использовании разработанной административной панели. На базе реализации сайта достигнуто поставленных целей: возможности регистрации и авторизации в системе, предоставление необходимой информации студентам-дипломникам, своевременный контроль и соблюдение графика выполнения дипломного проекта, быстрый и удобный способ поиска нужных материалов.

Ключевые слова: PHP, контент, сайт, админ панель, база данных, информационная система.

Lavrenchuk S.V., Hridin O.O. Information system for graduate and graduates of the department. This article describes how to create a responsive website for the Department of Computer Engineering which realized full functionality and the ability to use mobile devices. With plenty of benefits website enables easy management and monitoring of content, users, individual records using the developed administrative panel. On the basis of realization of the site achieved its goals: the possibility of registration and authorization system, providing the necessary information to students graduate, timely monitoring and compliance schedule of the diploma project, a fast and convenient way to find the right materials.

Keywords: PHP, content, website, admin panel, database, information system.

Постановка проблеми та огляд існуючих рішень. Широке застосування інформаційних технологій у різних сферах життя дозволяє значно підвищити ефективність роботи. У зв'язку з цим необхідними якостями сучасної людини стають високий рівень інформаційної культури, розвинутий інтелект, вміння грамотно працювати з інформацією, професіоналізм. Сучасний комп'ютер з підключенням до всесвітньої мережі Інтернет забезпечує доступ практично до всіх інформаційних надбань людства.

Інтенсивне навчання, що характеризується збільшенням обсягу навчального матеріалу та зменшенням часу засвоєння, потребує пошуку ефективних методів навчання, засобів контролю засвоєння знань, що значно підвищували б якість навчання студентів. Прагнення постійно оптимізувати навчально-виховний процес зумовило появу нових і вдосконалення використовуваних навчальних технологій різних рівнів.

Збільшення обсягу знань та обмеження часу для його викладання вимагає від сучасного викладача застосування ефективніших методів та технологій навчання. Розвиток комп'ютерних технологій дає можливості викладачеві та студентові спілкуватися дистанційно. Особливої уваги і консультацій потребують студенти-дипломники. Від того наскільки ефективно та вчасно керівник організує роботу таких студентів, надасть їм вичерпну інформацію організаційного характеру, залежить великою мірою успішне виконання та захист дипломної роботи чи проекту.

Метою роботи є швидке та зручне надання об'ємної інформації для студентів при виконанні ними дипломних робіт або проектів. Інформаційна система цікава для випускників кафедри тим, що містить корисну інформацію, методичні вказівки до дипломного проектування, оголошення, надає можливість обрати керівника, тощо. Розроблений програмний продукт також допомагає спростити ведення обліку та контролю студентів-дипломників, підтримувати зв'язок з випускниками кафедри.

Враховуючи специфіку розробки інформаційного веб-сайту було проаналізовано декілька навчальних веб-ресурсів і відповідно було виділено переваги та недоліки даних систем. Проаналізувавши знайдені прототипи можна сказати, що кожен веб-сайт по своєму забезпечує ресурси і викладає потрібну інформацію.

Для порівняння було розглянуто ряд інформаційних веб-ресурсів, призначених для інформування студентів різних спеціальностей які займаються дипломним проектуванням. Спочатку був розглянутий інформаційний сайт Харківського національного економічного університету, факультету «Економічної інформатики». До недоліків даного ресурсу можна віднести малу кількість додаткових мультимедійних елементів, на сайті присутні пункти, які є малофункціональними та незрозумілими для пересічного користувача, відсутня додаткова панель для відображення новин, а також не зовсім вдалий дизайн та верстка самого сайту. Перевагами даного інформаційного ресурсу є вдале розміщення області контенту завдяки чому легко сприймається розміщена інформація та розділи, в яких вона міститься, також використання бокової панелі для виділення основної інформації та корисних посилань є більш ніж вдалим.

Іншим прикладом є інформаційний веб-ресурс Державного хіміко-технологічного університету, кафедри «Інформаційних систем». До переваг даного веб-ресурсу можна віднести просту навігацію по контенту, та адаптивний дизайн. До недоліків можна віднести велику кількість додаткових віджетів. Головне меню занадто перевантажене категоріями, що ускладнює використання сайту.

Результати проведеного аналізу дозволяють зробити висновок про наявність недоліків та переваг при створенні веб-ресурсів.

Отже при розробці інформсаційної системи «Випускник» слід уникнути значних недоліків, а також реалізувати інформаційну систему з використанням обраних технологій розробки.

Завдання. Необхідно створити адаптивний сайт, який міг би інтегрувати в собі усю необхідну та корисну інформацію для дипломного проектування та використовуватися на мобільних пристроях; реалізувати на сайті можливості реєстрації та авторизації студентів для моніторингу відвідуваності сайту та для підтримки зв'язку з випускниками кафедри.

В роботі використовуються наступні інструменти: HTML5, CSS3, PHP, JavaScript і MySQL.

Виклад основного матеріалу й практичне значення результатів.

Для створення бази даних використано пакет OpenServer. Реалізована база даних, яка взаємодіє із системою, має на меті використання її викладачами кафедри або системним адміністратором для пошуку того чи іншого дипломника, наповнення ресурсу новою інформацією, внесення нових даних для збереження їх у електронній формі та відображення на головній сторінці сайту. Для зберігання усього контенту, який відображається на сайті, існує окрема функціональна база даних, якою зручно керувати з адміністративної панелі системи. Згідно базового скрипта існує певний захист сайту, студенти які не пройшли авторизацію чи реєстрацію скористатися даною системою просто не зможуть. Перед тим, як обговорювати та обирати інструментальні підходи до розробки та удасконалення процесу надання випускникам кафедри необхідної інформації для роботи з дипломним проектом, опишемо коротко послідовність дій дипломника. Як відомо, на початку навчального року студент-дипломник обирає собі керівника та разом із ним обговорює тему своєї дипломної роботи. Згодом, дотримуючись графіку виконання, він вивчає літературу в області даного питання, проводить аналіз аналогічних програм та систем і лише тоді починає розробляти свій власний продукт. Але разом із тим виникає проблема де можна дістати усю необхідну документацію, зразки, вимоги та правила оформлення для дипломної роботи. Саме тому інформаційна система (рис. 1), яка розробляється, має на меті заощадити час студента та викладача, за рахунок зручного доступу з будь якого комп'ютера, який підключений до локальної мережі.

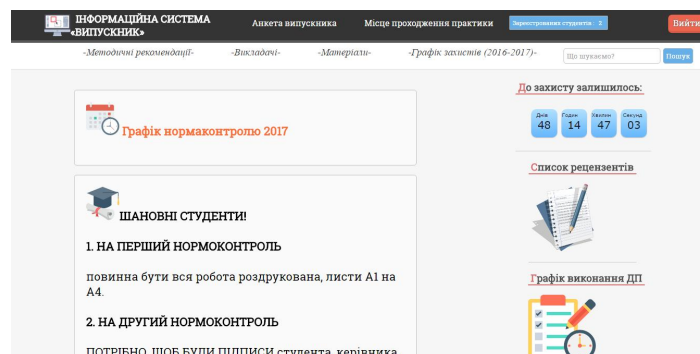


Рис. 1 – Головна сторінка сайту

На рис. 1 зображена головна сторінка сайту, URL на який посилається сайт – localhost/Intu.edu.ua. Структура даної сторінки наступна: головне меню; заголовок сайту; новини та оголошення; бокова панель; нижня частина. Уся інформаційна система реалізована мовою PHP.

На боковій панелі (рис. 2) присутні наступні елементи: таймер зворотнього відліку до початку захисту ДП; список рецензентів; графік виконання ДП; оформлення літератури; акт впровадження (зразок); вибір рецензента; вибір керівника ДП.

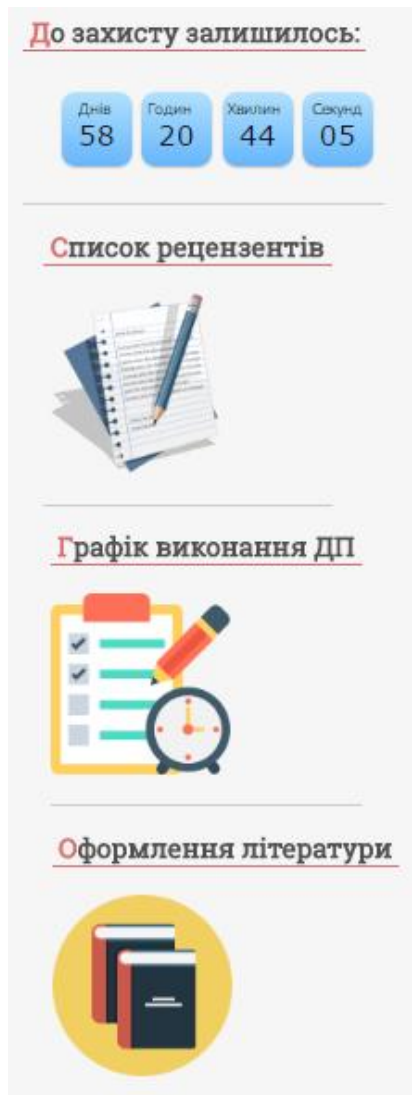


Рис. 2 – Бокова панель інформаційної системи

Реалізація поставлених завдань здійснюється за наступними кроками. Розробка системи була здійснена засобами PHP – 7. За допомогою розробленого скрипта, основу якого було взято із одного доступного Інтернет джерела [1], вдосконалено та спрямовано під реалізацію власної задачі. Це скрипт, який реалізує на сайті авторизацію та реєстрацію, а також за допомогою якого розроблена адмін-панель, приклад якої представлений на (рис. 3). Його функція полягає в тому, щоб кожен студент-дипломник міг зареєструватися в системі, виконати авторизацію та перейти на сайт, де він безпосередньо зможе отримати усю необхідну інформацію. Адміністративна панель, яка розроблена для зручного управління сайтом, призначена для адміністратора з метою контролю системи, видалення, редагування та додавання нової інформації.

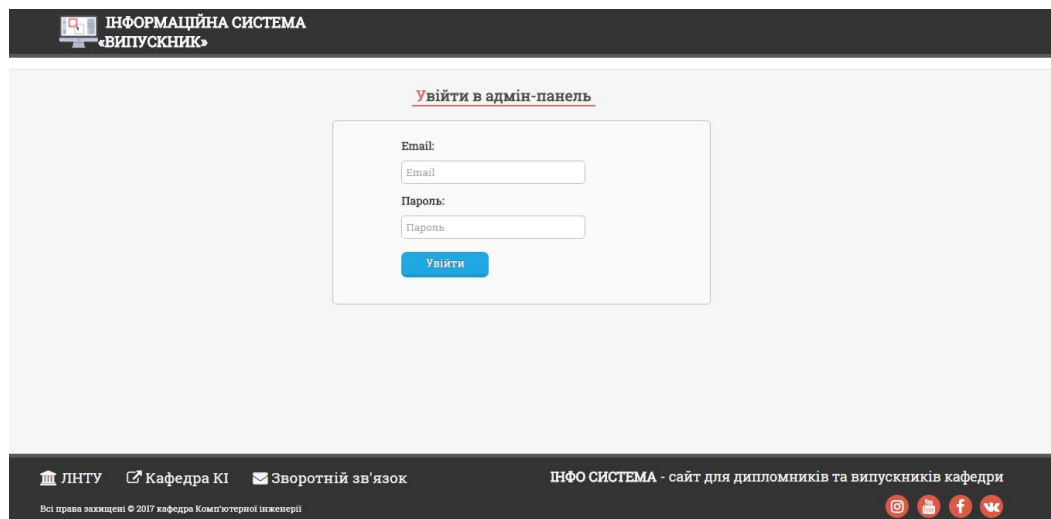


Рис. 3 – Адміністративна панель сайту

Для підвищення ефективності та надійності сайту було використано спеціальний скрипт з інтернет ресурсу [3], який дозволяє перевірити заповнені поля форми у два етапи. Перший етап полягає у тому, що перевірка здійснюється на стороні клієнта тим самим не перевантажуючи сервер. Даний скрипт використовує мову JavaScript та додатковий фреймворк jQuery. Другий етап перевірки відбувається на стороні сервера, так як злоумисник може вимкнути JavaScript у своєму браузері і тим самим потрапити на сайт. Тому другий етап є дуже важливий, він виконується за наступним принципом: якщо студент намагається авторизуватися у системі, але він у ній не зареєстрований то даний код порівнює введений email та пароль з вмістом бази даних і у випадку, коли він його не знаходить, користувач отримує повідомлення про те що його даних просто немає в системі. Також даний скрипт контролює аби студент не міг повторно зареєструватися, тобто щоб база даних не містила дублікати. Додатково реалізована перевірка на коректність введення email, номер телефону, та інших полів які повинні бути коректно заповненими.

Не менш важливою функцією, яка ставилась у пріоритетні завдання була реалізація на сайті форми зворотнього зв'язку (рис.4). Вона дає змогу студенту вести діалог з адміністратором в разі виникнення певних складнощів при користуванні системою або поставити питання щодо тематики дипломного проектування та інших можливих аспектів.

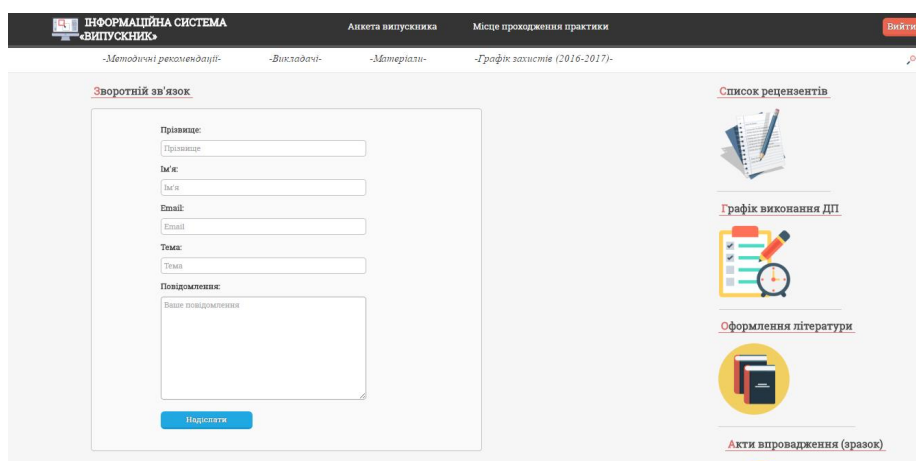


Рис. 4 – Зворотній зв'язок на сайті

Дана форма розроблена з використанням мови програмування PHP. Окрім базового функціоналу сайт містить додаткові функції зокрема це можливість швидко надати інформацію про місце проходження переддипломної практики (рис. 5), та заповнити анкету випускника. Після чого адміністратор зможе швидко отримати усі дані. Система, яка володіє додатковим функціоналом, дозволить заощадити час викладачів, на збір цієї інформації.

Рис. 5 – Місце проходження практики

Основну увагу під час розробки системи було приділено адаптивності, так як в подальшому існує ідея використання сайту на різних пристроях. Реалізація адаптивності (рис. 6), до різних типів екранів, відбувалася за наступними принципами: спочатку це було реалізовано та протестовано у браузері з відповідними налаштуваннями CSS файлів і додатково за допомогою фреймворка jQuery.

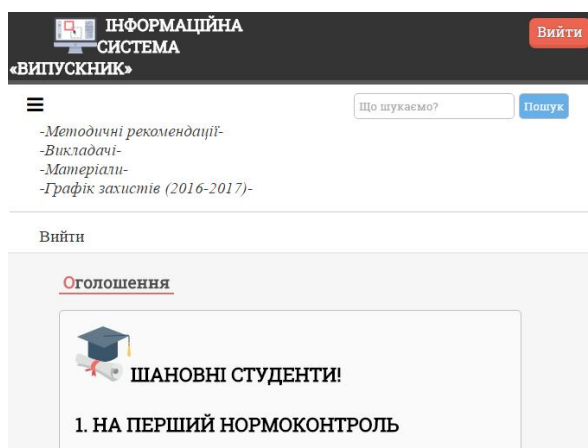


Рис. 6 – Адаптивність сайту

Висновок. Створений та випробуваний сайт для дипломників та випускників кафедри комп'ютерної інженерії є адаптивним для перегляду з різних пристроїв, має реалізацію ряду налаштувань та переваг, які дають можливість студенту-дипломнику зручно отримувати усю необхідну інформацію для виконання дипломно проектування. Досягнуто поставленої мети, а саме розроблено інформаційну систему, що містить повноцінний функціонал, який робить використання сайту ще простішим та зручнішим. Для випускника та дипломника тут зібрані усі матеріали, тому їх з легкістю можна переглянути, або завантажити. Важлива функція, яка реалізована в проєкті, це можливість вибору керівника та запропонованої теми дипломної роботи.

1. HTML CSS JavaScript jQuery HDOM AJAX [Електронний ресурс] – Режим доступу до ресурсу: <http://www.wisdomweb.ru/CSS3/>.
2. Пасічник В.В., Резніченко В.А., Організація баз даних та знань. – К.: Видавнича група BHV, 2006. – 384с.
3. Розроблення алгоритмів та методів завантаження даних інформаційної системи [Електронний ресурс] – Режим доступу до ресурсу: <http://vuzlib.com.ua/articles/book/38096.html>.
4. Peter Westwood. DataBase [Електронний ресурс] / Peter Westwood. – 2014. – Режим доступу до ресурсу: <http://www.citforum.ru/database/osbd/contents.shtml>.
5. Гаевский, А.Ю.; Романовский, В.А. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript; М.: Триумф, 2008. – 464 с.

УДК 004.254 (045)

Мельник В.М., Гаджемурра А.М.

Луцький національний технічний університет

ПЕРЕВАГИ ВИКОРИСТАННЯ СУБД SQLITE У ДОДАТКАХ WINDOWS

Мельник В. М., Гаджемурра А. М. Переваги використання СУБД SQLite у додатках Windows. В даній роботі розглядаються переваги системи управління базами даних SQLite для Windows-додатків. Заодно детально розглядаються і обґрунтовуються самі механізми в реалізації цих переваг. В роботі також пропонується приклад Windows-додатку із демонстрацією використання SQLite в інтеграції з засобами середовища розробки Qt Creator та мови програмування високого рівня C++. Описані також способи підключення баз даних та основні операції запису даних і читання їх з бази.

Ключові слова: SQLite-переваги, система управління базами даних, середовище розробки Qt Creator, мова високого рівня C++, Windows-додатки.

Мельник В. М., Гаджемурра А. М. Преимущества использования СУБД SQLite в приложениях Windows. В данной работе рассматриваются преимущества системы управления базами данных SQLite для Windows-приложений. Совместно рассматриваются и обосновываются сами механизмы в реализации этих преимуществ. В работе также предлагается пример Windows-приложения с демонстрацией использования SQLite в интеграции со средствами среды разработки Qt Creator и языка программирования высокого уровня C++. Описанные также способы подключения баз данных и основные операции записи данных и считывания их с базы.

Ключевые слова: SQLite-преимущества, система управления базами данных, среда разработки Qt Creator, язык высокого уровня C++, Windows-приложения.

Melnyk V. M., Gadzhemurra A. M. Advantages in DBCS SQLite using for Windows applications. This paper considers the advantages of the database management system SQLite for Windows applications. There are jointly considered and justified mechanisms in these advantages implementation. The paper also offers an example of a Windows application that demonstrates the use of SQLite in integration with the development environment Qt Creator and the high-level programming language C++. Also ways of databases connecting and the basic operations of data writing and reading them from the database are described.

Keywords: SQLite-advantages, database management system, development environment Qt Creator, high-level language C++, Windows applications.

Вступ

Великі об'єми інформації та необхідність ними керувати призвели до створення баз даних та систем управління ними (СУБД). На базі розробок з залученням баз даних розгортаються цілі клієнт-серверні системи. Однак у невеликих додатках, де обчислювальні ресурси досить обмежені *актуально* використовувати не такі вже і складні бази даних. СУБД SQLite являє собою полегшену реляційну систему управління базами даних, яка може бути втілена у вигляді бібліотеки, де реалізовано багато принципів і механізмів зі стандарту SQL-92 [1]. Серцевинний код SQLite поширюється як суспільне надбання, тобто може використовуватися без обмежень та безоплатно з будь-якою метою [2].

Дана СУБД дуже поширена для додатків у середовищах операційних систем Android, Windows, Linux та UNIX. Проте *проблематикою даного напрямку* у наш час є досить не значна кількість прикладів та описів для застосування у підручниках, посібниках чи іншій навчальній літературі. З особливістю це прослідковується у середовищі розробки Qt Creator, яке в Україні не має високої популярності.

Теоретичні відомості

Сама бібліотека SQLite написана мовою C, а її особливістю є те, що вона не використовує парадигму клієнт-сервер. Це говорить про те, що рушій SQLite не є окремим процесом, з яким взаємодіє додаток, а він тільки надає бібліотеку, з якою програма компілюється і рушій стає складовою частиною програми. Таким чином, як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite. Даний підхід зменшує накладні витрати, час відгуку і значно спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується додаток. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції весь файл, який зберігає базу даних, блокується. ACID-функції (Atomicity – атомарність, Consistency – узгодженість, Isolation – ізолюваність, Durability – довговічність) досягаються зокрема за рахунок створення файла-журналу. Кілька процесів або потоків можуть одночасно з легкістю читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, коли жодних інших запитів на

запис у цей час не обслуговується, інакше спроба запису закінчується невдачею і в програму-додаток повертається код помилки [3].

Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу. У комплекті постачання йде також функціональна клієнтська частина у вигляді файлу виконання `sqlite3`, за допомогою якого демонструється реалізація функцій основної бібліотеки. Клієнтська частина працює з командного рядка, і дозволяє звертатися до файлу бази даних на основі типових функцій операційної системи. Завдяки архітектурі рушія можливо використовувати `SQLite` як на вбудованих системах, так і на виділених машинах з гігабайтними масивами даних.

Метою роботи послужила також розробка нескладного додатку у створенні та заповненні бази даних із демонстрацією можливостей СУБД `SQLite` як наведеного зразка. Сама програма буде являти собою `Organizer` із попередньо заданим спливаючим повідомленням. Дослідження в даному напрямку потребують великих удосконалень та ефективних універсальних програм для роботи з СУБД.

Виклад основного матеріалу

`SQLite` – це реляційна база даних. Поняття реляційності пов'язане з розробками відомого англійського спеціаліста в області систем баз даних Едгара Кодда[4]. Ця модель характеризується простотою структури даних, зручним для користувача табличним представленням і можливістю використання формального апарату алгебри відношень і реляційного обчислення для обробки даних. Реляційна модель орієнтована на організацію у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і володіє наступними властивостями:

- кожен елемент таблиці являє собою один елемент даних;
- всі комірки в стовпці таблиці однорідні, тобто всі елементи в стовпці мають однаковий тип;
- кожен стовбець має унікальне ім'я;
- однакові рядки в таблиці повинні бути відсутні;
- порядок наступності рядків і стовбців може бути довільним.

Для того, щоб представити інформацію, що міститься в таблиці бази даних, у середовищі `Qt Creator` використовується кілька класів:

- `QSqlQueryModel` – це клас моделі, яка формує таблицю шляхом задавання сирого `SQL`-запиту. Вона може бути корисна при формуванні особливо витончених фільтрів і компіляції інформації з різних таблиць бази даних. Про неї докладніше можна отримати інформацію в.
- `QSqlTableModel` – це клас-основа нашого обговорення в даній статті. Дана модель формує таблицю за іменем тієї таблиці, яка існує в базі даних. Як недолік можна відзначити відсутність методів підключення зв'язків з іншими таблицями для того, щоб підставляти значення в поля з інших таблиць через параметр `ID`.
- `QSqlRelationalTableModel` – це клас, який дозволяє формувати таблицю зі зв'язками з іншими таблицями, підміняючи значення таблиці, яку представляє дана модель, через `ID`-записи, що містяться в інших таблицях.

Функціональні можливості додатка

Модуль `QtSql` використовує плагіни драйверів для взаємодії з `API` для різних баз даних. Так як `API SQL` модуля не залежить від баз даних, то код, специфічний для певної бази даних, міститься в її драйверах. Деякі драйвери поставляються разом з `Qt`, а інші можуть бути додані певним шляхом. Перш, ніж отримати доступ до бази даних за допомогою класів `QSqlQuery` або `QSqlQueryModel`, ми повинні встановити з базою даних хоча б одне з'єднання. З'єднання з базою даних ідентифікуються за допомогою довільних рядків. `QSqlDatabase` також підтримує концепцію з'єднання за замовчуванням, яке використовується класом `Qt SQL` за замовчуванням, якщо ніяке інше з'єднання не вказано. Цей механізм дуже зручний для додатків, що використовують тільки одне з'єднання з базою даних [5].

Приклад коду, що встановлює з'єднання з базою даних `SQL` з використанням додаткових параметрів може бути представлений наступним чином:

```
CInfo::CInfo() :  
    db(QSqlDatabase::addDatabase("QSQLITE"))  
{  
    db.setHostName("localhost");  
    db.setDatabaseName("myBase");
```

```
db.setUserName("root");
db.setPassword("pass");
db.open();
db.setDatabaseName("job.db");
bIsOpened = db.open();
if(bIsOpened)
{
    updateDBVersion();
}
}
```

QSqlDatabase::addDatabase() – це ім'я драйвера. Для отримання списку драйверів, дивіться документацію addDatabase() [8]. Для ініціалізації даних з'єднання необхідно викликати ряд функцій setHostName(), setDatabaseName(), setUserName() і setPassword(). Для видалення з'єднання з базою даних спочатку слід закрити базу даних за допомогою власного метода QSqlDatabase::close (), а потім потрібно видалити її за допомогою статичного методу QSqlDatabase::removeDatabase()[6].

Для виконання запитів SQL необхідно створити об'єкт QSqlQuery і викликати метод QSqlQuery::exec(). QSqlQuery надає одноразовий доступ до результуючої бази вибіркою одного запиту. Після виклику exec() внутрішній покажчик QSqlQuery вказує на позицію перед першим записом. В цьому випадку необхідно викликати метод QSqlQuery::next() один раз, щоб перемістити покажчик на перший запис таблиці. Потім знову необхідно повторювати виклик метода next(), щоб отримувати доступ до інших записів до тих пір, поки він не поверне значення false[7].

Функція QSqlQuery::value() повертає значення поля поточного запису. Поля задаються індексами, починаючи з нуля. Функція QSqlQuery::value() повертає значення типу QVariant, який може зберігати значення різних типів C++ і ядра Qt, такі навіть як int, QString і QByteArray. Різні типи значень бази даних в Qt автоматично приводяться до найближчого еквівалента. Можна переміщатися вперед і назад по вибірці, використовуючи функції QSqlQuery::next(), QSqlQuery::previous(), QSqlQuery::first(), QSqlQuery::last() і QSqlQuery::seek(). Поточний номер рядка можна отримати за допомогою функції QSqlQuery::at(), а загальна кількість рядків у вибірці, якщо це підтримується базою даних, повертається функцією QSqlQuery::size().

У наведеному нижче прикладі вказувати з'єднання не потрібно, а тому використовується з'єднання за замовчуванням.

```
QSqlQuery query;
query.exec ("SELECT * FROM tabMain;");
while (query.next ())
{
    tableWidget-> insertRow (0);
    tableWidget-> setItem (0, 0, new QTableWidgetItem (query.value (0) .toString ());
    tableWidget-> setItem (0, 1, new QTableWidgetItem (query.value (1) .toDate () .toString ());
    tableWidget-> setItem (0, 2, new QTableWidgetItem (query.value (2) .toString ());
    tableWidget-> setItem (0, 3, new QTableWidgetItem (query.value (3) .toString ());
    tableWidget-> setRowHeight (0, 20);
}
```

Якщо виникає помилка, метод exec() повертає значення false. Доступ до помилки можна отримати за допомогою функції QSqlQuery::lastError().

Вставляючи запис в таблицю, тобто використовуючи INSERT, можна одночасно вставити багато записів. Але найчастіше ефективніше відокремити запит від реально вставлених значень. Це можна зробити за допомогою вставки значень через параметри[8]. У наступному прикладі показані вставки записів за допомогою проіменованого параметра:

```
QSqlQuery query;
query.prepare("INSERT INTO tabMain VALUES (null,: datetime,: string,: int);");
query.bindValue(": datetime", QDateTime::currentDateTime ());
query.bindValue(": string", lineEdit-> text ());
query.bindValue(": int", spinBox-> value ());
query.exec ();
```

Крім зручності виконання, вставка через параметри має ще й ту перевагу, що розробник позбавлений від необхідності піклуватися про перетворення спеціальних символів. Зміна записів дуже схожа на вставку в таблицю:

```
QSqlQuery query;  
query.prepare("UPDATE tabMain SET String = ': string', Int =: int WHERE ID =: id;");  
query.bindValue(": string", lineEdit-> text ());  
query.bindValue(": int", spinBox-> value ());  
query.bindValue(": id", iID);  
query.exec ();
```

Також наведемо приклад виразу REMOVE:

```
QSqlQuery query;  
query.prepare ( "DELETE FROM tabMain WHERE ID =: id;");  
query.bindValue(": id", tableWidget-> item (tableWidget-> currentIndex (). row (), 0) -> text());  
query.exec ();
```

Повна функція зчитування усіх даних з бази виглядатиме так:

```
bool CInfo::getLastRowData(int &id, QDateTime &dt)  
{  
    QString query("SELECT max(id) FROM jobs;");  
    QSqlQuery q = db.exec(query);  
    bool result = q.lastError().type() == QSqlError::NoError;  
    if(result)  
    {  
        id = q.record().value(0).toInt();  
        query = "SELECT date_created FROM jobs WHERE id = " + QString::number(id) + ";";  
        q = db.exec(query);  
        result = q.lastError().type() == QSqlError::NoError;  
        if(result)  
        {  
            dt = QDateTime::fromTime_t(q.record().value(0).toUInt());  
        }  
    }  
    return result;  
}
```

Особливості функціонування додатку

Результат роботи додатка Organizer демонструє єдине значення (рис. 1). При натисненні кнопки "Edit" можна побачити усю інформацію про запис з можливістю її редагування. У даному випадку це дані рядкового типу та типу дата/час, які при необхідності можна змінювати. Усі наведені вище програмні коди знаходяться у файлі-ресурсі mainwindow.cpp

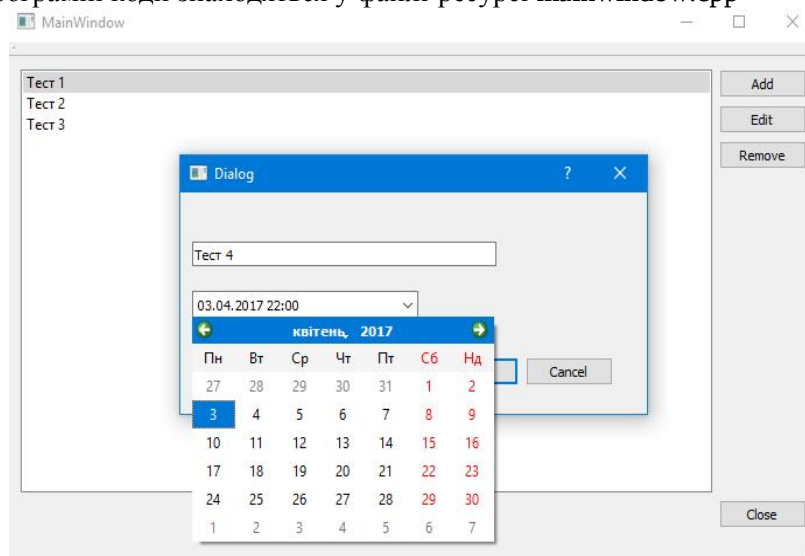


Рис. 1 – Результат роботи додатка

У диспетчері задач можна переглянути ресурсозатратність для операційної системи Windows 10, яка для прикладу наведена на рисунку 2.



Процес	Використання процесора	Використання пам'яті	Використання мережі	Використання диска
Wid.exe (32 біт)	0%	22,2 МБ	0 Мбіт/с	0 Мбіт/с
MainWindow				

Рис. 2 – Використання ресурсів комп'ютера розробленим додатком

Споріднені роботи

Застосування SQLite разом із середовищем Qt Cretor охоплює досить широке коло додатків завдяки вільному його використанню. Проте науково-дослідних робіт, які б досліджували конкретно цю СУБД, порівняно мало. Її використовують разом із готовими програмними додатками, де важлива загальна продуктивність усієї системи. Дана стаття наочно продемонструвала ефективність вищеназваних засобів. У науковій статті «Захист SQL-бази в операційній системі Android» за авторством О. М. Гайдаєнко вказано за якими стандартами шифрувати дані у базі для їх криптостійкості. [9] Така інформація може доповнювати поточний додаток, розроблений у даній статті потрібно лише розробити необхідний алгоритм шифрування.

Інша робота «Розробка бази даних інформаційної системи діагностики глаукоми» [10]. База даних там із великою кількістю таблиць та зв'язків, проте не вказано за яким принципом буде надаватись доступ до цих даних, відповідно швидкодія може бути вкрай низькою, якщо не оптимізувати код.

Висновки. На основі проведених досліджень, які передбачались науково-дослідною роботою, проведений аналіз характеристик використання СУБД SQLite та відзначено ряд переваг для, які дозволяють зрозуміти її потенціальні переваги і в подальшому використовувати їх у програмних додатках, створених для операційної системи Windows. Наведений варіант розробленої програми-дodatка, яка може застосовуватись як Organizer для нагадування про певну подію, використовуючи при цьому мінімум системних вимог та ресурсів. Сформований алгоритм може послужити як приклад застосування СУБД SQLite при створенні та управлінні найрізноманітніших баз даних для Windows-додатків з урахуванням зауважених переваг, що значно спрощуватиме роботу для розробників програмного забезпечення.

1. С.Д. Кузнецов. Стандарты языка реляционных баз данных SQL: краткий обзор. / SIT Forum. Новости мира IT. http://citforum.ru/database/articles/art_2.shtml
2. А.Д. Хомоненко. База даних: учебное пособие. / Хоменко А.Д., Цыганков В.М., Мальцев М.Г. – ООО «КОРОНА-Век» 190005, Санкт-Петербург. – 736 с.
3. Саак А.Э. Информационные технологии управления: учебник для вузов / А.Э. Саак, Е.В. Пахомов, В.Н. Тюшняков. – СПб.: Питер, 3-е изд., 2011. – 640 с.
4. Д. Елджер. Библиотека программиста C++, – К.: Грамота.
5. Qt Downloads. <https://www.qt.io/>
6. SQLite Home Page. <http://www.sqlite.org/>
7. С. Кузнецов. Понятие модели данных. Обзор разновидностей моделей данных. / SIT Forum. Новости мира IT. http://citforum.ru/database/advanced_intro/6.shtml.
8. SqlQueryModel Class. QT Documentation. <http://doc.qt.io/qt-5/qsqlquerymodel.html>.
9. О. М. Гайдаєнко. - «Захист SQL-бази в операційній системі Android». – Вінницький Національний Технічний Університет, Вінниця.
10. О. В. Висоцька, І.Ю. Панф'орова, Г.М. Страшенно, С.О. Синенко, Ю.А. Дьомін. – «Розробка бази даних інформаційної системи діагностики глаукоми». – Харківський Національний Університет, Харків.

UDK 004.254 (045)

Мельник В.М., Мельник К.В., Багнюк Н.В., Гринюк С.В.

Луцький національний технічний університет

ANDROID-BUILT CODE GENERATION MODELLING FOR HETEROGENEOUS ARCHITECTURE

Melnyk V., Melnyk K., Bagniuk N., Hryniuk S. Android-built code generation modelling for heterogeneous architecture. The success of Android is based on its unified Java programming model that allows to write platform-independent programs for a variety of other target platforms. However, this comes at the cost of performance. As a consequence, Google introduced APIs that allow to write native applications and to exploit multiple cores as well as embedded GPUs for compute-intensive parts. Here is proposed code generation techniques in order to target the Renderscript and Filterscript APIs. Render script harnesses multi-core CPUs and unified shade GPUs, while the more restricted Filterscript also supports GPUs with earlier shade models. Our techniques are focused on image processing applications that allow to target these APIs and OpenCL from a common description. We further supersede memory transfers by sharing the same memory region among different processing elements on HSA platforms. As reference, we use an embedded platform hosting a multi-core ARM CPU and an ARM Mali GPU. We show that our generated source code is faster than native implementations in OpenCV as well as the pre-implemented script intrinsic provided by Google for acceleration on the embedded GPU.

Мельник В.М., Мельник К.В., Багнюк Н.В., Гринюк С.В. Модель генерації вбудованого Андройд-коду для гетерогенної архітектури. Переваги Android базуються на його уніфікованій моделі програмування Java, що дозволяє писати незалежні від платформи програми для різних цільових платформ. Проте це відбувається за рахунок зниження продуктивності. Як наслідок, Google представив API-інтерфейси, що дозволяють писати власні програми та використовувати кілька ядер, а також вбудовані графічні процесори для ресурсомістких частин. В статті пропонується методи генерації коду з метою виявлення API, Renderscript і Filterscript. Renderscript використовує багатоядерні процесори і графічні процесори уніфікованого відтинку, в той час як більш обмежена Filterscript також підтримує графічні процесори з більш ранніми тінювими моделями. Наша технологія орієнтована на додатки для обробки зображень, які дозволяють призначати подібні інтерфейси і OpenCL з загального опису. Крім того, здійснюється витіснення перекладів пам'яті, розділяючи одну і ту ж область пам'яті між різними елементами обробки на HSA-платформах. В якості порівняння використовується вбудована хостинг-платформа багатоядерного процесора ARM і ARM Mali GPU. Продемонстровано, що згенерований вихідний код є швидшим, ніж власна реалізація в OpenCV, а також попередньо реалізований сценарій, характерно наданий Google для прискорення на вбудованому GPU.

Introduction

The steady desire for new applications and higher display resolutions drives the development of embedded platforms and the need for faster, more powerful processors at the same time. Today's mobile platforms found in smartphones and tablets host multi-core Central Processing Units (CPUs) and even programmable embedded Graphics Processing Units (GPUs) to deliver the demanded performance. This raises the question how to harness the processing power of these parallel and heterogeneous platforms. As a remedy, Google proposed two new parallel programming concepts for Android that allow to target CPUs as well as GPUs: Renderscript and Filterscript.

These programming models were designed for the predominant application domain of image processing with portability in mind. While applications in Android use Java as programming language, Renderscript and Filterscript are based on C99. Hence, Java programmers have to write low-level C code in order to benefit from the high performance that is provided by these new programming models. This work proposes code generators that allow to automatically generate Renderscript and Filterscript code. The generated target code is derived from a Domain-Specific Language (DSL) for image processing algorithms [1]. The proposed code generators for Renderscript and Filterscript are based on the existing compiler infrastructure, which provides back ends for CUDA and OpenCL on discrete, standalone GPU accelerators.

The focus in this work is on the code generation that allows to target the different components in today's heterogeneous embedded platforms.

We present the first code generator for Renderscript and Filterscript on Android platforms starting from an abstract high-level representation. The generated implementations are even faster compared to the target-specific implementations in the *Open Source Computer Vision* (OpenCV) framework. At the moment, the

algorithm description is compact and requires only a fraction compared to available highly optimized implementations.

The target code is generating for embedded *Heterogeneous System Architecture* (HSA) platforms. With HSA, CPU and GPU share the same physical memory (Fig. 1). This allows us to avoid extensive memory transfers and enables the employment of heterogeneous resources where the same data has to be accessed frequently from different compute resources.

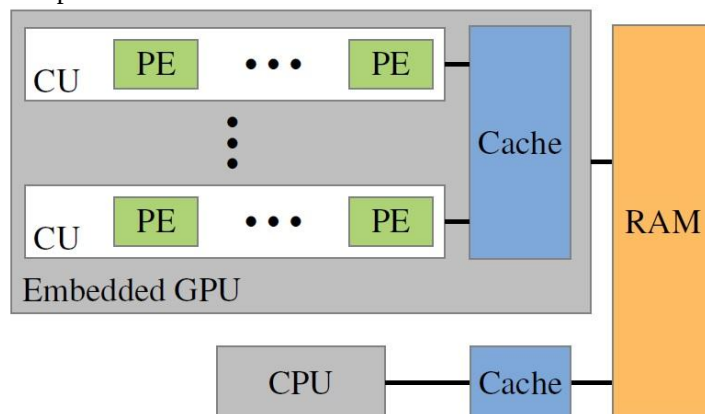


Fig. 1. Structure of typical HSA platform with an embedded GPU

Fig. 1: Structure of a typical HSA platform with an embedded GPU

Programming models on embedded devices

The Android operating system is widespread on end user mobile devices but it becomes more and more interesting to use it also on other embedded devices as found in the industrial automation or the automotive sector. For example, compute-intensive advanced driver assistance systems such as self- and convoy-driving technology, parking guidance, and better infotainment systems are becoming more and more important for automotive.

For such compute-intensive tasks, energy efficient embedded GPUs are of particular interest. However, there exists no common programming language that allows to harness also embedded GPUs. Android, for example, provides multiple programming models that allow to target a huge variety of devices with different usage scenarios in mind:

a) SDK: The Android Software Development Kit (SDK) is based on the Java programming language. The SDK source code is compiled to bytecode that is executed by the Dalvik V.M., which is either directly interpreting the bytecode or compiling it into machine-specific instructions and executing it. Hardware-specific characteristics are hidden and not exposed to the programmer. For that reason it is not possible to gain benefit from certain hardware specific optimizations like vectorization. This dramatically limits the overall achievable performance.

b) NDK: The Android Native Development Kit (NDK) promises much better execution performance. Even though complete Android applications can be developed using C/C++ in the NDK, native code is usually executed by an application that is written using the SDK through the Java Native Interface (JNI).

Using the NDK, hardware-specific characteristics are transparent to the programmer and can be utilized, for instance, by compiler intrinsics. For efficiently supporting a wide range of different CPU architectures, native code needs to be rewritten beforehand in a way that exploits particular architecture features to achieve high performance. For that reason, usually only compute-intensive code segments are realized in native parts of an Android application.

c) Renderscript: Google first introduced the Renderscript programming language in 2011. The aim of this new language is to provide a programming model that avoids performance issues of the SDK without introducing portability problems the NDK suffers from. Renderscript is based on C99 and provides additional support for vector types. The Renderscript front end compiler generates an intermediate representation which

is then further compiled to native code that is optimized for a specific available target architecture like CPUs, Digital Signal Processors (DSPs), and GPUs. Unlike other parallel computing Application Programming Interfaces (APIs) such as OpenCL or CUDA, Renderscript was not designed with performance as the primary goal. Hardware-specific features like local memory are hidden from the programmer and, hence, target-specific optimizations like tiling may not be exploited.

The number of threads is directly inferred from the output buffer where each element is processed by a single thread. Another limitation is that the actual execution target (CPU, GPU, DSP) cannot be specified and is automatically chosen by the runtime system. These limitations ensure portability at the expense of absolute performance.

d) Filterscript: Filterscript is a subset of Renderscript with certain limitations and stricter constraints to ensure a wider compatibility among CPUs, GPUs, and DSPs. Filterscript files are used and compiled the same way as Renderscript.

The major difference is that pointers are not allowed. Therefore, memory cannot be directly read using linear arrays. Instead, the provided access API functions must be used. Only gather reads are supported, which means that only one output value can be written per thread (in contrast to scatter writes). Instead of assigning the value to a buffer, it is returned by the kernel function and written to the global ID of the current thread. Further limitations are relaxed floating point precision and lack of 64-bit built-in types.

e) OpenCL: The Open Compute Language (OpenCL) programming model is not officially supported on Android devices. However, on recent devices, like the Nexus 4 smartphone and the Nexus 10 tablet, a working OpenCL driver can be found within the system libraries. Note that Google pulled the unofficial OpenCL support in Android 4.3.

Providing a well-known common API like OpenCL cannot hide the fact that embedded GPU architectures vary considerably from their desktop counterparts. For instance, the local memory is not dedicated on-chip memory, but is instead mapped to global memory. Another major difference is that compute cores of most embedded GPUs are Very Long Instruction Word (VLIW) architectures and gain a huge benefit from vectorization or the use of vector types.

OpenCL supports the allocation of host accessible memory using `map()` and `unmap()`. On desktop GPUs, this feature can be used to define page-locked host memory, which may speed up memory transfers. On HSA platforms like the ARM Mali [2] and AMD Fusion [3], these operations are used to avoid copying buffers. Because the main memory is shared among CPU and GPU, both can access the same memory region without copies.

Target code generation

This section introduces the target code generation for the parallel computing APIs introduced in Section II. We use a DSL for image processing as basis for target code generation and employ source-to-source translation to target different parallel programming models.

The Heterogeneous Image Processing Acceleration (HIPA^{cc}) Framework

The DSL provided by the HIPA^{cc} framework [1] is based on C++ and provides built-in classes for two-dimensional images and other objects such as filter masks. An image in the DSL stores the image pixels and can be initialized from plain C data arrays:

```
uchar *image = readPGM(&width, &height, "lena.pgm");  
Image<uchar> in(width, height);  
in = image;
```

Similarly, a filter mask can store the stencil used for a convolution:

```
const float filter_mask[] =  
{  
    0.057118f, 0.124758f, 0.057118f,  
    0.124758f, 0.272496f, 0.124758f,  
    0.057118f, 0.124758f, 0.057118f  
}  
Mask<float> mask(3, 3);  
mask = filter_mask;
```

Using these data abstraction classes, computations on multidimensional image objects can be defined. A computational kernel is defined as a C++ class that holds a kernel() method describing the computation on a single pixel. Within the kernel method, all memory accesses are relative to the current pixel and only members of the C++ class can be accessed. Considering the Gaussian blur filter as an example, its computation can be expressed using relative memory accesses to the mask filter mask and the input image. The result is written back using the output() method:

```
void kernel()
{
    float sum = 0;
    int range = size/2;
    for (int yf = -range; yf <= range; ++yf)
        for (int xf = -range; xf <= range; ++xf)
            sum += mask(xf, yf) * input(xf, yf);
    output() = (uchar) sum;
}
```

A more concise and expressive syntax for common computational patterns such as convolutions are provided by the HIPA^{cc} DSL as well: the convolve() method describes the convolution of an image with a mask:

```
void kernel()
{
    output() = convolve(mask, SUM, [&] () -> float {
        return mask() * input(mask); });
}
```

Using the convolve method is not only more compact, it gives the source-to-source compiler also more freedom to optimize the code.

The HIPA^{cc} compiler [1] generates target CUDA and OpenCL code from programs written in this DSL. Using source-to-source translation, instances of DSL C++ classes are replaced by corresponding API calls to the CUDA and OpenCL runtime library provided by HIPA^{cc}. Compute kernels, in contrast, are not mapped one-to-one to corresponding CUDA and OpenCL. Instead, the Abstract Syntax Tree (AST) of a kernel is analyzed and optimizations are applied such as staging image pixels into local memory or mapping multiple iterations to one GPU thread (loop unrolling).

Memory accesses are then redirected to memory fetches from global memory, texture memory, or local memory—depending on the target device. Similarly, Mask accesses get mapped to constant memory or propagated as constants in case the operator is described using the convolve() function.

Renderscript and Filterscript Support for HIPA^{cc}

We have extended HIPA^{cc} for Renderscript and Filterscript support, adding a new back end for each API. Program parts of the DSL responsible for resource management are mapped to corresponding commands in the runtime library, which we provide. The compute-intensive kernels, however, are translated into Renderscript and Filterscript kernels. These get initialized at program start and can be executed afterwards.

1) Memory Access Mapping: As highlighted in Section II, memory accesses are handled differently in OpenCL, Renderscript, and Filterscript. Therefore, the introduced back ends map reads and writes to an Image to corresponding API calls and memory array accesses. To illustrate this, we consider a simple read from an Image, followed by a write in HIPA^{cc}:

```
Image<uchar> input;
...
void kernel()
{
    uchar val = input();
    output() = val;
}
```

In OpenCL, these memory accesses are mapped to 1D memory arrays that are added to the signature of the kernel function with corresponding attributes indicating that the arrays reside in global CPU or GPU memory. In case neighboring pixels are read, the x and y index is adjusted accordingly:

```
__kernel void kernel(__global const uchar *input,  
__global uchar *output, ...)  
{  
    uchar val = input[y*width + x];  
    output[y*width + x] = val;  
}
```

Renderscript provides data buffers for storing image data (rs_allocation) and rsGetElementAt API calls for reading/writing data elements. Rather than writing the result to the current iteration point (i. e., writing to the first kernel parameter), the result is stored to _iter, the currently processed pixel:

```
rs_allocation input;  
...  
void kernel(uchar *_iter, uint32_t x, uint32_t y)  
{  
    uchar val = rsGetElementAt_uchar(input, x, y);  
    *_iter = val;  
}
```

In Filterscript, data buffers are defined and read as in Renderscript, but no API calls are provided for storing results. Instead, the result for the current thread (pixel) is returned by the kernel:

```
rs_allocation input;  
...  
uchar __attribute__((kernel)) kernel(uint32_t x,  
uint32_t y)  
{  
    uchar val = rsGetElementAt_uchar(input, x, y);  
    return val;  
}
```

In order to map the execution of a kernel in Renderscript and Filterscript either to the CPU or to the GPU, environment variables are used.

In order to achieve high performance, the kernel has to be mapped to the memory hierarchy of the target architecture. GPUs provide multiple memory types apart from the global memory that are optimized for different access patterns such as constant memory, texture memory, or local memory. OpenCL allows to explicitly use these memory types in the source program. For example, filter masks are typically mapped to constant memory and read-only images with high spatial or temporal locality to texture memory or local memory. In contrast to this, Renderscript and Filterscript do not support any explicit mapping to the memory hierarchy.

2) *Iteration Space Mapping*: HIPA^{cc} allows the programmer to define a Region of Interest (ROI) in the output image to be computed. Similarly, only an ROI on input images can be read. This allows to work on images of different size in one kernel and to process only image regions of interest. The ROI on the output image defines also the iteration space and the number of threads required for kernel execution. This iteration space size is used as launch configuration in parallel compute APIs such as CUDA and OpenCL and offsets to the image are passed to the kernel in order to process only the ROI. However, the native API of Renderscript and Filterscript does not provide launch configurations up to Android 4.4. Instead, the buffer holding the image data defines also the launch configuration. That is, for each pixel in the image a thread is started resulting in an index space that is larger than the iteration space defined by the programmer.

To overcome this deficiency, there exist three approaches. First, we can define a buffer with the size of the iteration space. This temporary buffer stores the result of the kernel and is copied back to the ROI in the output image as specified by the programmer. This approach requires additional memory of $ROI_{width} \cdot ROI_{height}$ and requires one additional memory transfer of the same size. Second, we can define a dummy

buffer with dimensions equal to the iteration space and use this buffer to provide the index space. Result pixels are not stored to this buffer, but to the buffer associated with the output image of the kernel. This approach requires in theory no additional memory and no additional memory transfers, but can only be used for Renderscript. However, it turns out that the Renderscript runtime still allocates memory for the buffer although no data is associated with the buffer. In Filterscript, the output pixel is not written to a buffer, but returned within the kernel. Third, we can use an index space with a size of the whole output image and add guards to the kernel so that only threads of the index space calculate pixels that are also part of the iteration space. This launches $(\text{IMG}_{\text{width}} \cdot \text{IMG}_{\text{height}}) - (\text{ROI}_{\text{width}} \cdot \text{ROI}_{\text{height}})$ additional threads that do not compute pixels. While this approach is valid for Renderscript, the behavior is undefined in Filterscript in case no return statement is executed. Hence, we read the corresponding pixel value from the output image for index points outside of the iteration space and return those. This requires $(\text{IMG}_{\text{width}} \cdot \text{IMG}_{\text{height}}) - (\text{ROI}_{\text{width}} \cdot \text{ROI}_{\text{height}})$ additional memory reads and writes for the Filterscript implementation, but has no memory allocation overhead. This approach is the only one that provides a valid iteration space mapping for Filterscript with only little overhead. Thus, this approach is followed by the code generator in this work.

3) *Vector Support*: HIPA^{cc} supports scalar data types for discrete GPUs from AMD and NVIDIA. These GPUs schedule scalar instructions to single lanes of their Single Instruction, Multiple Data (SIMD)-like architecture. Adjacent threads are mapped to adjacent lanes. The embedded CPUs and GPUs considered here require vector instructions, though. Otherwise only a fraction of peak performance can be achieved. Therefore, we add vector type support to HIPA^{cc}, which is compatible with the syntax in OpenCL and Renderscript.

Support for HSA Memory Management

The HIPA^{cc} DSL was designed for desktop systems and therefore has no particular support for HSA platforms. Device memory is abstracted from the developer by the Image class in HIPA^{cc}. Memory transfers to the device are handled implicitly by the framework. On HSA targets, it is possible to share memory between CPU and GPU by using host accessible memory that must be allocated using OpenCL API calls.

We added support for HSA platforms to HIPA^{cc} by extending its memory management to abstract host memory as well and implicitly manage map() and unmap() operations. This additional abstraction has the benefit that a) faster page-locked memory can be utilized on desktop systems, and b) the same memory region can be used for the CPU and GPU on HSA platforms in order to avoid memory copies. In case memory is allocated by third party frameworks (e. g., OpenCV or FreeImage), the programmer can still manage host memory explicitly.

Related works

While there exist a wide range of frameworks and compilers that generate low-level assembly code for embedded architectures, there is only little related work on targeting the new compute APIs Renderscript and Filterscript.

For example, the Portland Group introduced the PGCL framework [6], which adds support for OpenCL to Android on the ST-Ericsson NovaThor platform. Their compiler supports the NEON instruction set and vectorization for ARM multi-core CPUs. Halide [7], a DSL for image processing provides a back end for ARM NEON, and the OpenCV [8] library utilizes also the ARM NEON instruction set. These frameworks have all in common to target only the CPU. More recently, a source-to-source translator for mapping annotated Java code (using pragmas similar to OpenMP within source code comments) to Renderscript and OpenCL was presented [9]. Hence, the developer requires profound knowledge of OpenMP, while our domain-specific approach abstracts from low-level architecture details.

Qian, Zhu, and Li [10] compare and analyze the programming model of the SDK, NDK, and Renderscript considering usability aspects such as programmability, but also performance. They conclude that Renderscript provides the best performance while preserving portability at the cost of manual memory management and difficult library extensibility. The solution presented in this work does not have these limitations since the Renderscript code and the supporting files are automatically generated. Moreover, we support also Filterscript and OpenCL as back ends, which allows to map program parts to the GPU and CPU.

GMAC [11] provides similar memory management abstractions for discrete GPUs (CUDA, OpenCL), but do not consider HSA platforms. However, the proposed abstractions can also be integrated into GMAC.

Table II: Lines of code for the OpenCV implementation, HIPA^{cc} DSL code and generated Renderscript code.

	Sobel	Gaussian	Laplace	FIR	Harris
OpenCV	1681	1641	1712	982	2247
HIPA ^{cc} DSL	15	22	11	11	68
Renderscript	1915	1951	8575	3680	4265

Conclusion

A code generator model has been presented for Renderscript and Filterscript in the domain of image processing by utilizing the HIPA^{cc} DSL. All code variants should be automatically generated from a common description that is highly portable and performs well on both: CPU and GPU. The code generator has to be integrated into the HIPA^{cc} framework and is available as open-source under <http://hipacc-lang.org>.

Using this code generator, we'll try to produce the efficient code in our next research for different parallel programming models on embedded devices. We also predict to perform better than the target-optimized OpenCV library on the CPU.

References

1. R. Membarth, F. Hannig, J. Teich, M. Körner, W. Eckert, "Generating Device-specific GPU Code for Local Operators in Medical Imaging", in International Parallel & Distributed Processing Symposium (IPDPS), IEEE, May 2012, pp. 569–581.
2. ARM. (2013). Mali-T600 Series GPU OpenCL – Developer Guide.
3. B. A. Hechtman, D. J. Sorin, "Evaluating Cache Coherent Shared Virtual Memory for Heterogeneous Multicore Chips", in International Symposium on Performance Analysis of Systems and Software (ISPASS), Jun. 2013.
4. Arndale Board, Samsung Exynos 5 Dual Arndale Board, <http://www.arndaleboard.org>, 2012–2013.
5. C. Harris and M. Stephens, "A Combined Corner and Edge Detector", in Alvey Vision Conference, 1988, pp. 147–151.
6. The Portland Group, PGI OpenCL Compiler for ARM, <http://www.pgroup.com/products/pgcl.htm>, 2011–2013.
7. J. Ragan-Kelley, A. Adams, S. Paris, M. Levoy, S. Amarasinghe, F. Durand, "Decoupling Algorithms from Schedules for Easy Optimization of Image Processing Pipelines", ACM Transactions on Graphics (TOG), vol. 31, no. 4, 32:1–32:12, Jul. 2012.
8. Willow Garage, Open Source Computer Vision (OpenCV), <http://opencv.willowgarage.com/wiki>, 1999–2013.
9. Acosta, F. Almeida, "Towards a Unified Heterogeneous Development Model in Android", in International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms (HeteroPar), Springer, Aug. 2013.
10. X. Qian, G. Zhu, X.-F. Li, "Comparison and Analysis of the Three Programming Models in Google Android", in Asia-Pacific Programming Languages and Compilers Workshop (APPLC), ACM, Beijing, China, Jun. 2012, pp. 1–9.
11. Gelado, J. E. Stone, J. Cabezas, S. Patel, N. Navarro, W.-M.W. Hwu. "An Asymmetric Distributed Shared Memory Model for Heterogeneous Parallel Systems", in International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), ACM, Mar. 2010, pp. 347–358.

УДК 514.18

Міскевич О. І., Кокоша М. С.

Луцький національний технічний університет

ДОДАТОК «СИСТЕМНИЙ МОНІТОР» ЗАСОБАМИ БІБЛІОТЕКИ QT.

Міскевич О. І., Кокоша М. С. Додаток «Системний монітор» засобами бібліотеки Qt. Розроблено додаток для моніторингу системних ресурсів (обсяг спожитої пам'яті, файлу підкачки, завантаженість процесора, примонтовані файлові системи і т.п.) засобами Qt під ОС Linux. Дана програма може служити для користувачів невеликою утилітою для відслідковування стану їх системи.

Ключові слова: Qt, Linux, відкрите програмне забезпечення, системний монітор, диспетчер задач, системне програмне забезпечення

Міскевич О. И., Кокоша М. С. Приложение «Системный монитор» средствами библиотеки Qt. Разработано приложение для мониторинга системных ресурсов (объем потребленной памяти, файла подкачки, загрузка процессора, смонтированные файловые системы и т.п.) средствами Qt под ОС Linux. Данная программа может служить для пользователей небольшой утилитой для отслеживания состояния их системы.

Ключевые слова: Qt, Linux, открытое программное обеспечение, системный монитор, диспетчер задач, системное программное обеспечение.

Miskevych O. I., Kokosha M.S. Application «System Monitor» base on Qt Library. Developed application for monitoring of system resources (like as amount of consumed memory, swap memory, mounted file systems, etc) base on Qt library for Linux OS. This application is a small utility for users that want to monitor performance state of own system.

Keywords: Qt, Linux, open-source software, system monitor, task manager, system software.

Вступ. Продуктивність – це один з найбільш помітних параметрів системи. Із збільшенням кількості ресурсоемного програмного забезпечення часто постає проблема у її нестачі. Для відслідковування стану системи користувачами чи адміністраторами призначені утиліти і програмне забезпечення, що служать для визначення характеристик заліза комп'ютера, його тестування і моніторингу стану основних компонентів в реальному часі. У даній статті здійснюється короткий огляд існуючих засобів для моніторингу системи, можливостей бібліотеки Qt; операційної системи Linux, а також приділяється увага розробці додатку "Системний монітор".

Постановка проблеми. У ході роботи було поставлено завдання створити додаток для моніторингу стану основних параметрів системи, що працюватиме під ОС Linux. Для розробки додатка використано мову програмування C++ та середовище розробки Qt Creator.

Аналіз існуючих рішень. Більшість дистрибутивів Linux поставляється з великим числом програмного забезпечення та утиліт, які призначені для моніторингу системних ресурсів і продуктивності.

До цього числа можна включити такі основні консольні інструменти:

- **top** (консольна команда, що виводить список виконуваних процесів і основну інформацію про них), а також деякі її модифікації – **atop**, **htop**.
- **iotop** – утиліта, подібна до утиліти **top**, але відображає використання не CPU і пам'яті, а роботи процесів з дисками, написана на Python. Допоможе визначити який процес звертається до жорсткого диска в Linux. Відображає активні процеси, які в даний момент виконують операції запису / читання з диску, збирає статистику за певний час.
- **powertop** – утиліта, що дозволяє виявити в системі компоненти, які споживають більше енергії, ніж потрібно і показує загальне електроспоживання (у Вт), інформація зчитується з різних джерел ядра. Це дозволяє експериментувати з налаштуваннями для керування електроживленням, ефективно налаштувати споживання енергії під вашу машину.
- **iftop** – виводить інформацію про активні мережеві з'єднання, швидкість мережевого завантаження / віддачі, моніторить онлайн-трафік, розділяє трафік за протоколами, інтерфейсами і хостами.
- **vnStat** – утиліта для обліку мережевого трафіку, зберігає історію мережевого трафіку для обраних інтерфейсів. **vnStat** отримує дані з ядра Linux.

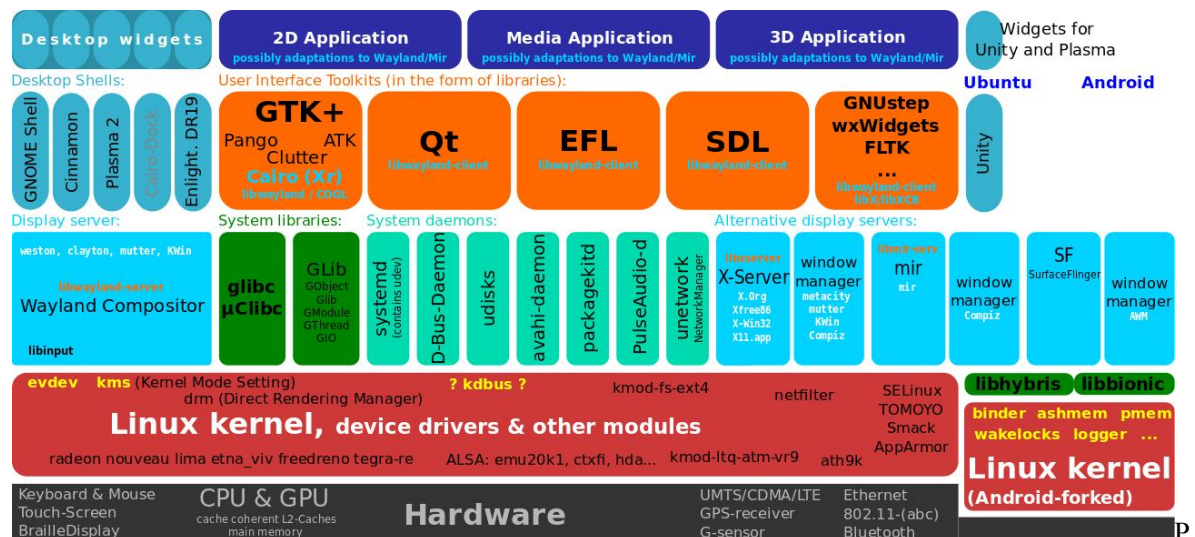
А також графічні інструменти:

- Gnome system monitor – утиліта виводить у вигляді графіків інформацію в реальному часі про ресурси - використання процесора (CPU), використання оперативної пам'яті (RAM) і файлу підкачки (SWAP), а також використання мережі.
- KDE System Guard – менеджер завдань та монітор ресурсів і продуктивності, що є частиною стільничного середовища KDE. Фактично має ті самі функції, що gnome system monitor. Написано на C++ (Qt).
- Psumon – зручний графічний додаток для перегляду поточних процесів і моніторингу стану системи. Написано мовою Python (Qt). Psumon має простий користувальницький інтерфейс і мінімум налаштувань, дозволяє в режимі реального часу стежити за використанням системних ресурсів і системними процесами з можливістю їх примусового завершення.
- Conky – вільна програма для моніторингу стану системи для X Window System. Характерною ознакою Conky є надзвичайна гнучкість в налаштуваннях, що дозволяє кожному користувачеві підлаштувати її інтерфейс та вивід системної інформації за своїми уподобаннями. На відміну від інших програм для системного моніторингу, які використовують спеціальні складні віджети, Conky відображається напряму у вікні X Window. Це дозволяє використовувати менше системних ресурсів у порівнянні з іншими програмами.

Це далеко не всі програми для моніторингу системних ресурсів, представлені у світі open-source, їх набагато більше. Для наглядності на рис. 1 зображено open-source інструментарії та бібліотеки для розробки серверного, мережевого ПЗ та програмного забезпечення із графічним інтерфейсом користувача.

Як бачимо з даної схеми Qt займає чільне місце серед основних інструментаріїв для розробки ПЗ з графічним інтерфейсом користувача. Ще одним із таких фреймворків є GTK+ - кросплатформовий набір інструментів для створення графічних інтерфейсів користувача. Разом із Qt є одним із найпопулярніших інструментів для X Window System. Gnome system monitor (GSM), згаданий вище, написаний за допомогою цього інструментарію.

У власному додатку було використано середовище Qt. Додаток за своїм функціоналом є облегшеною версією Gnome system monitor.



рисунки 1 - Open-source інструментарії для розробки ПЗ

Виклад основного матеріалу роботи. Робота додатку "Системний монітор" по-суті базується на парсингу деяких підкаталогів віртуальної файлової системи *proc* та використанні стандартних бібліотек та засобів C++ і бібліотеки Qt. Для кращого розуміння необхідно познайомитися з */proc* ближче.

Файлова система `/proc` є механізмом для ядра і його модулів, що дозволяє надсилати інформацію процесам (звідси і назва `/proc`). За допомогою цієї віртуальної файлової системи можна працювати з внутрішніми структурами ядра, отримувати корисну інформацію про процеси і змінювати конфігурацію (змінюючи параметри ядра) на льоту. За допомогою файлів в `/proc` можна отримати інформацію про стан ядра, процесів, параметрах комп'ютера і т.д. Більшість файлів в `/proc` містять найсвіжішу інформацію про системне обладнання. Незважаючи на те, що ці файли віртуальні - їх можна переглянути будь-яким текстовим редактором або за допомогою команд "more", "less" або "cat". [1] Перелічимо деякі важливі файли:

- `/proc/cpuinfo` - інформація про процесор (модель, сімейство, розмір кеша і т.д.).
- `/proc/meminfo` - інформація про RAM, розмір файлу підкачки і т.д.
- `/proc/mounts` - список змонтованих файлових систем.
- `/proc/devices` - список пристроїв.
- `/proc/filesystems` - підтримувані файлові системи.
- `/proc/modules` - список завантажуваних модулів.
- `/proc/version` - версія ядра.
- `/proc/cmdline` - список параметрів, переданих ядру при завантаженні. [2]

Насправді файлів в каталозі `/proc` набагато більше. Можна детально переглянути всі файли в `/proc` командами "more" і read. Для прикладу наведу функцію, що отримує загальний процесорний час із каталогу `/proc/stat`:

```
unsigned long long getTotalCpuTime()
{
    FILE* file = fopen("/proc/stat", "r");
    if (file == NULL) {
        perror("Неможливо зчитати файл");
        return 0;
    }
    char buffer[1024];
    unsigned long long user = 0, nice = 0, system = 0, idle = 0;
    unsigned long long iowait = 0, irq = 0, softirq = 0, steal = 0, guest = 0, guestnice = 0;
    char* ret = fgets(buffer, sizeof(buffer) - 1, file);
    if (ret == NULL) {
        perror("Неможливо зчитати файл");
        fclose(file);
        return 0;
    }
    fclose(file);
    sscanf(buffer,
        "cpu %16llu %16llu %16llu %16llu %16llu %16llu %16llu %16llu %16llu %16llu",
        &user, &nice, &system, &idle, &iowait, &irq, &softirq, &steal, &guest, &guestnice);
    // http://unix.stackexchange.com/q/178045/20626
    return user + nice + system + idle + iowait + irq + softirq + steal;
}
```

Так як для розробки було вибрано бібліотеку Qt, то давайте дещо ознайомимося з нею.

Qt – багатоплатформовий інструментарій розробки ПЗ на мові програмування C++. Є також «прив'язки» до багатьох інших мов програмування: Python - PyQt; Ruby - QtRuby ; Java - Qt Jambi та інші.

З часу своєї появи в 1996 році бібліотека Qt лягла в основу тисяч успішних проєктів по всьому світу. Крім того, Qt є фундаментом популярного робочого середовища KDE, що входить до складу багатьох дистрибутивів Linux.

Відмітна особливість Qt від інших бібліотек — використання Meta Object Compiler (МОС) — попередньої системи обробки вихідного коду. МОС дозволяє у багато разів збільшити потужність бібліотек, вводячи такі поняття, як слоти і сигнали. Утиліта МОС шукає в заголовних файлах на C++ опису класів, що містять макрос Q_OBJECT, і створює додатковий вихідний файл на C++ , що містить метаоб'єктний код [3].

Перелічимо основні переваги Qt над іншими інструментаріями:

- Використання сигналів і слотів замість функцій зворотнього виклику, що значно спрощує розуміння коду і робить його лаконічним
- Qt Assistant — детальна довідкова система Qt, що спрощує роботу з документацією по бібліотеці.
- Відсутність потреби слідкувати вручну за виділенням пам'яті — у Qt при видаленні класу-предка усі класи-нащадки знищуються автоматично.
- Можливість написання кросплатформенного коду — треба лише скомпілювати код під платформую, на якій працюватиме програма.

Отож, перейдемо до UML діаграми класів додатку (рис. 3). Вона показує структуру класів програми — її атрибути і методи, а також взаємозв'язки між об'єктами самих класів.

Пройдемося по призначенню основних класів програми.

Класи ProcessInformationWorker, resourcesWorker, fileSystemWorker — фактично є каркасами трьох вкладок додатку — процеси, пам'ять і файлові системи (рис. 2)

Класу processPropertiesDialogue відповідає діалогове вікно, що містить таблицю із більш детальними властивостями вибраного процесу.

Клас workerThread відповідає за управління потоками для кожної із 3-х вкладок додатку і діалогових вікон.

Клас MainWindow відповідає за будову головного вікна програми, старт і своєчасне видалення класів потоків при виході з програми чи при раптовому її зупиненні, створення головного меню та виклик діалогового вікна.

Класи TableMemoryItem та TableNumberItem відповідають за коректне порівняння значень колонок у таблицях.

Класи QObject, QMainWindow, QTableWidgetItem, QTableWidgetItem, QDialog, QAction, QCheckBox та інші — рідні класи бібліотеки Qt, призначені для програмування графічного інтерфейсу користувача (детальніше про їх методи та властивості можна дізнатися у довідковій системі Qt Assistant).

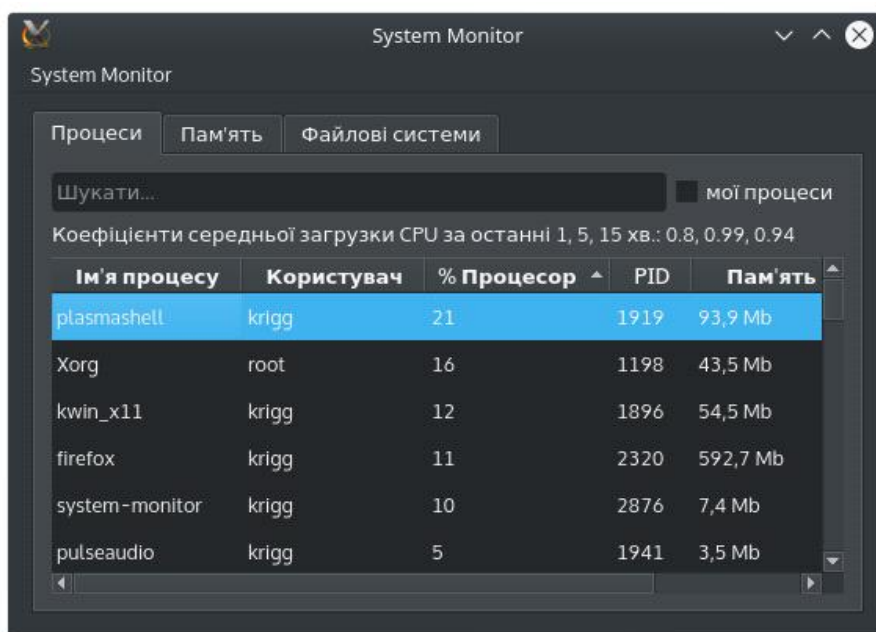


Рисунок 2 - Загальний вигляд головного вікна програми



Рисунок 3 — UML діаграма класів додатку

Висновки. В результаті проведеного дослідження була створена невелика утиліта — системний монітор, що виводить список виконуваних процесів системи із можливістю їх завершення, також надає інформацію про примонтовані файлові системи і пам'ять.

Крім того перелічено основні вільні аналоги даного додатку, визначено переваги використання бібліотеки Qt для розробки власного ПЗ. Також пояснено призначення основних класів додатку і наведено їх UML діаграму. Варто також відмітити, що даний прототип системної утиліти можна, по можливості, оновлювати і нарощувати функціонал. Щодо подальшого вдосконалення додатку та його функціоналу, то пізніше можна зреалізувати наступне:

- наповнити додаток графіками, що інформуватимуть в реальному часі про продуктивність системи;
- додати вкладку “Мережі”, що дозволить переглядати активні мережеві з'єднання, швидкість мережевого завантаження / віддачі, моніторити переданий трафік;
- додати вкладку "Інформація про систему", що сповіщатиме нас про апаратне забезпечення комп'ютера.

1. А. Соловьев. Разработка модулей ядра Linux (kernel linux module). - 2011. - С. 85
2. Файловая система /proc. Режим доступа: <http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref-guide/ch-proc.html>
3. Шлее М. Профессиональное программирование на C++. Qt 5.3. БХВ-Петербург. - 2015. - С. 923
4. <https://doc.qt.io>
5. <http://stackoverflow.com/>
6. <https://en.wikipedia.org>

УДК 004.58

Міскевич О. І., асистент, Коренюк А.Ю.

Луцький національний технічний університет

РОЛЬ ЛЮДИНИ В ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЯХ

Міскевич О. І., Коренюк А.Ю. Роль людини в інформаційних технологіях. Стаття пояснює роль людей які підпорядковуються реаліям сучасного світу. Розглядаються способи виключення взаємодії користувача з живою особою. Також розглядається способи формування інноваційних технологій в сучасному світі.

Ключові слова: ресурс, інформаційні технології, людська праця, веб-застосунки.

Міскевич О. И., Коренюк А.Ю. Роль человека в информационных технологиях. Статья объясняет роль людей, которые подчиняются реалиям современного мира. Рассматриваются способы исключения взаимодействия пользователя с живой личностью. Также рассматривается способы формирования инновационных технологий в современном мире.

Ключевые слова: ресурс, информационные технологии, человеческий труд, веб-приложения.

Miskevich O. I., Koreniuk A. Y. The role of man in information technology. The article explains the role of people who are subject to the realities of the modern world. The methods of exclusion user interaction with a live person. We also consider ways of forming innovative technologies in the world today.

Keywords: resources, information technology, human labor, Web applications.

Вступ. З періоду створення перших комп'ютерів, людство не розуміло скільки користі може принести звичайний калькулятор.

На даний момент, в часи інформаційних технологій, прогрес не стоїть на місці, хоча і можна побачити спад активності розробки нових і нових систем. Основними критеріями порівняння виключень роботи людей є заміна живої праці на роботизовані системи, або так звані «штучні інтелекти».

Постановка наукової проблеми. Метою даної роботи є аналіз існуючих способів заміни людей програмованими системами на основі веб-ресурсів, та їх подальшого втілення як повна автоматизація праці на підприємствах, офісах, магазинах, тощо. Розглянути основні принципи взаємодії з не органічними працівниками.

Аналіз існуючих способів.

Згідно з Законом України «Про інноваційну діяльність» [1], інновації це новостворені або вдосконалені конкурентоздатні технології, продукція або послуги, а також організаційно-технічні рішення виробничого, адміністративного, комерційного або іншого характеру, що істотно поліпшують структуру та якість виробництва та соціальної сфери.

Звичайно застосовується соціальна інновація якщо розглядати в сфері роботи підприємств, адже це сприяє покращенню життя населення, хоча в ній є більше мінусів ніж плюсів.

При використанні автоматизованих робочих станцій замість персоналу, робота та праця в цілому не є затратною. Розробник який створить програмне забезпечення по обслуговуванню клієнтів отримає винагороду за працю, але це позбавить робочого місця продавця, касира, сортувальника, тощо.

Зазначимо те, що у такий спосіб роботодавець набере значно менше персоналу, але оплата праці буде вища. Зрозуміло, що ми не можемо гарантувати якість обслуговування на вищому рівні, але для цього і є варіанти відповідей на питання або ж додаткове підключення живої людини в спілкування, хоча це не є вигідно в рамках економічного процесу підприємства.

Суттєвою фрустрацією є те, що у такий спосіб, коли клієнт вибираючи своє замовлення, він не бачить товар який хоче придбати, це якщо розглядати інтернет-магазин. Сучасні ресурси переходять на нову ступінь еволюції продажу який розглянеться в подальших прикладах.

Розглянемо принцип взаємодії технологій минулих років і сучасності.

Маючи в кишені декілька грошових одиниць, людина має можливість обрати річ, яка їй до вподоби, але залишається тільки знайти те, що підпаде під певні критерії. Вибираючи навіть кофтину з фіксованою ціною, покупець може змінити ціну поговоривши з продавцем, але це не головний аспект, а те що стан товару можна оцінити і детально ознайомитись зі всіма доступними компонентами та документами перед самою покупкою. Саме цей спосіб продажів завоював ринок тогочасних 80-90х років минулого століття, хоча і займала велику частку часу у людей.

На даному етапі розвитку технологій ця практика заміняється з виходом інтернету в вільний доступ, та саме головне що доступність інформації ніяк не обмежується крім деяких пунктів заборони по віку або статі, також розглядається расова нерівність та інші приклади фільтру людей.

Щоб придбати телевізор необхідно спочатку вибрати магазин в який піти, потрібну модель, її розмір, функціональні характеристики, простоту використання, цінову політику та саме головне працездатність. І так людство жило десятиліттями, а телевізор це простий приклад.

Сьогодні нам дає можливість зайти на звичайний сайт, який написаний людиною і вибрати те, що потрібно. Але не все так просто як здається, ось з цього місця детальніше.

Для роботи інтернет-магазину потрібен веб-ресурс, але щоб цей ресурс створити, потрібна людина яка знає як він створюється, людина в той же час має знати якісь певні навички в розробці таких ресурсів, саме тому в сучасному світі стає популярним комп'ютерна інженерія, програмування та різні сфери інформаційних технологій, серед яких найбільший попит мають дизайн, веб розробка, та взагалі розробка програмного забезпечення так як програмне забезпечення має основу яка кріпиться на апаратне забезпечення, яке застосовується в цей же час для розробки різного устаткування. Розробка цього устаткування має багато переваг, тому що затрати на розробку інтернет магазину для прикладу, будуть менші ніж створення цього магазину в умовах середовища типу в місті, тому що є ціна за оренду, світло, опалення, оплату роботи персоналу, доставка товару та інші чинники працевлаштування осіб, тощо.

Розробнику для роботи потрібна мова програмування яка була створена можливо ще до його народження так як сучасні технології освоюють з дитинства. А деякі мови програмування були описані ще декілька років тому, і вони є в вільному доступі та підтримуються багатьма редакторами. З їх допомогою можна створити оригінальний дизайн, оригінальну ідею та застосувати його у своєму подальшому розвитку.

Чому саме інтернет-магазин? Нікому не секрет, що в нашому житті керують гроші і виходить так, щоб створити інтернет магазин де йде повна заміна робочої сили на технології сучасного світу типу повна автоматизація з моменту виробництва товару до моменту отримання його людиною.

Написання певного коду для того щоб звичайно створити веб-ресурс не потрібно, це може зробити і старшокласник.

Хоча принцип роботи включає і людську форму діяльності але необхідність людей ніхто не виключає до того моменту коли штучний інтелект буде розумніший ніж людський мозок. Але це вже біологічний розвиток людини який не відноситься до цієї теми, тому як людська праця в майбутньому буде замінена повністю на робочу зону комп'ютерів за підтримки програмного забезпечення яке буде розроблятися людьми, що робить нас вимираючим видом як якась тварина яку просто знищує сфера діяльності яку ми ж самі розробляємо. Можна сказати що людина створює ставок, запускає туди рибу але збираючи її вона не піклуються про потомство вона знищує рибу яку виростила і так виживає хоча технологіям не потрібне виживання, вони харчується енергію яку людство навчилось видобувати ще десятки років тому.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Чому саме людина стає вимираючим видом? А ви просто задумайтесь як можна придбати річ у сучасному магазині.

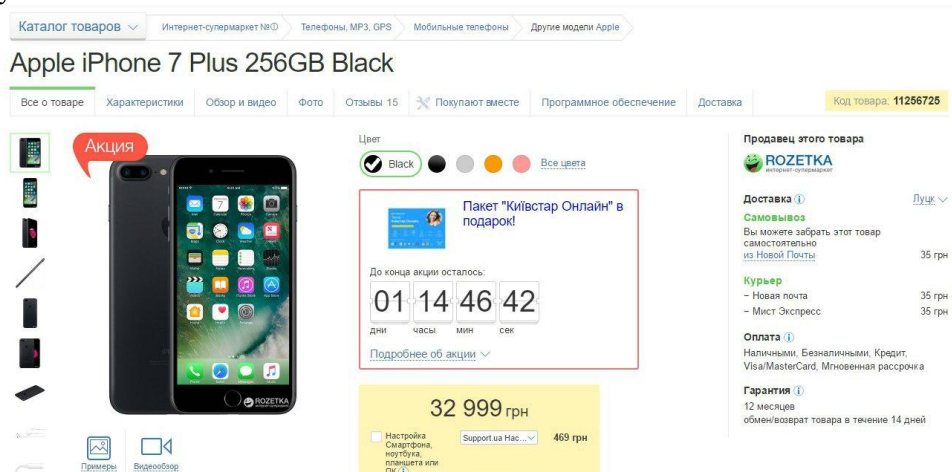


Рис. 1. Сучасна купівля товарів

Заходиш на ресурс який був розроблений під потреби користувача, авторизуєшся, хоча це не завжди потрібно, вибираєш за певними критеріями річ яку ти навіть не бачив, не можеш відчутти на дотик, і довіряєш продавцю який рекламує товар, це якщо брати за основу інтернет магазини. Це не є суттєвим недоліком сучасних реалій, а саме те, що заміна людей на ЕОМ які значно витриваліші розумніші швидші та більш функціональні ніж людина. Саме тому ця тема є актуальною на сьогоднішній день.

Можна говорити про те, що людина яка керується грошима, може створити такі ресурси та обманювати інших людей тому, що один чоловік який може створити декілька таких ресурсів може заподіяти дуже велику шкоду, та щоб знайти його, потрібно докласти багато зусиль так як самі Люди створюють програмне забезпечення яке захищає їх від пливу інформаційних технологій, тобто для того щоб знайти людину яка відправила повідомлення потрібно дуже багато зусиль, хоча в більшості випадків таку людину знайти неможливо тому процвітає інтернет шахрайство та інформація приховується навіть від професійних шукачів злочинців.



Рис. 2. Вигляд коду веб-ресурсу

Є досить багато фільмів в яких розповідається про захоплення світу штучним інтелектом, і на сьогодні людство боїться штучного інтелекту так як він може стати розумнішим ніж сама людина

Вже є докази того що штучний інтелект читаючи деякі веб-ресурси, а точніше інформації яка знаходиться на них, створюють перші вірші, звичайно низької якості, але це вже значний крок в сферу заміни машини замість поетів. Навіщо тоді поет, якщо машина створить в десятки разів більше віршів ніж людина, швидше, якісніше та споживає менше ресурсів для створення чогось.

На сьогоднішній день актуальність пристроїв, так званих сучасних технологій, комп'ютерів телефонів, планшетів і таке інше, значно більша ніж людей для людей в минулих десятиріччях. На кожну людину в світі припадає від трьох пристроїв на кожну. Хоча є люди які відмовляються від технологій загалом, і ніколи ними не користуються за все своє життя, але на відміну від людей які застосовують у побуті сучасні технології вони відстають в розвитку

Вченими доказано, що діти які розвиваються з одного року за допомогою комп'ютерів планшетів та телефонів, розвиток відбувається значно швидше ніж звичайні діти які розвиваються наприклад від книжок або просто слухаючи своїх батьків. З дитинства нам нав'язують що дитина повинна вчитися від технологій, але це правда, і результати тестів це підтверджують.

Що буде якщо забрати людини сучасні технології такі як комп'ютери, планшети, телефони та іншу техніку, або взагалі вимкнути світло. Без світла людина не зможе довго існувати, адже вона звикла до всіх приладів та інновацій. Але цивілізація у ХХІ столітті не може існувати без цих технологій які зараз захоплюють світ, і можливо через декілька років, такі статті як ця, ніколи не будуть публікуватися тому, що читати буде нікому.

Висновки

Керуючись матеріалами статті можна вважати, що полегшення праці за допомогою технологій вбиває людину як біологічний вид. Прямий зв'язок цього є те, що для того щоб зробити

щось корисне можна знайти як це зробити в всесвітній мережі інтернет, але також саме полегшення сприяє розвитку людини.

В людей є доступ до всієї інформації світу, але вони використовують лише 10% всієї доступної інформації, тому, що 80% інформації яка знаходиться в інтернеті є неправдивою або придуманою, яка немає підтвердження, але на неї найбільше звертають увагу. Також є 10% які залишилися і це та інформація яка потрібна людині але доступ до неї обмежений або її важко знайти.

Використовуючи «всесвітню павутину» людина шукає звичайний «рецепт борщу» чи наприклад «як зав'язати галстук», а не для тих цілей яких він був створений, і це реальний приклад деградації людства.

Спосіб який базується на використанні комп'ютера для взаємодії з іншими людьми є гарним прикладом того, що інформація має новий спосіб передачі між людьми, хоча не тільки люди використовують пристрої, тварини також можуть використовувати, наприклад спеціальні кормушки з сенсором підходячи до якої, тварина отримує їжу або інші послуги.

Те, що вважається як інформації, як технологія і застосовуються в сільському господарстві приносить користь, але кожна розробка має свою ціну, для когось ціна в грошах а для нас вона в житті.

1. Закон України Про інноваційну діяльність від 4 липня 2002 р., № 40 - IV.
2. Коутс Р., Влеймінк І. Інтерфейс «людина комп'ютер». - Москва.: «Мир», 1990. - 501с.
3. Поспелов Д.А. Інтелектуальні інтерфейси для ЕОМ нових поколінь. Режим доступу: www.raai.org/about/persons/pospelov/pages/interf.doc.
4. Курзанцева Л.І. Про адаптивному інтелектуальному інтерфейсі «Користувач - система масового застосування Інтерфейс «людина комп'ютер» // Комп'ютерні засоби, мережі та системи: Зб. Наук. Праць.- Київ: Ін-т кібернетики ім. В.М. Глушкова НАНУ.- 2008.- №7.- С. 110-116.
5. Курзанцева Л.І. Про застосування агентної технології при створенні інтелектуального призначеного для користувача інтерфейсу // Комп'ютерні засоби, мережі та системи: Зб. Наук. Праць.-Київ: Ін-т кібернетики ім. В.М. Глушкова НАНУ.- 2003.- №2.- С. 15-24

УДК 62.50:658.21

Пастернак Я.М., Братанов А.С.

Луцький національний технічний університет

ПОРІВНЯЛЬНИЙ АНАЛІЗ ТИПОВИХ СИСТЕМ КЕРУВАННЯ КОНТЕНТОМ (CMS)

Пастернак Я.М., Братанов А.С. Порівняльний аналіз типових систем керування контентом (CMS). У даній статті було проведено аналіз п'яти найрозповсюдженіших систем керування контентом. Наведено можливості кожної системи, вказано критерії оптимального використання кожної з них. Всі системи є абсолютно безкоштовними, а також мають відкритий код.

Ключові слова: контент, система, система управління контентом (вмістом).

Пастернак Я.М., Братанов А.С. Сравнительный анализ типичных систем управления контентом (CMS). В данной статье был проведен анализ пяти самых распространенных систем управления контентом. Приведены возможности каждой системы, указаны критерии оптимального использования каждой из них. Все системы являются абсолютно бесплатными, а также имеют открытый код.

Ключевые слова: контент, система, система управления контентом (содержимым).

Pasternak Ia.M., Bratanov A.S. Comparative analysis of typical content management system (CMS). This paper analyzes five most common content management systems. It is shows main capabilities of each system, and criteria for optima usage of each of them are presetted. All systems are completely free and open source.

Keywords: content, system, content management system.

Вступ. Системи керування контентом (Content Management System, або CMS) істотно спрощують вирішення поставлених задач веб-розробникам і навіть замовникам. На сьогоднішній день вони швидко розвиваються, але в їх проектуванні та розробці повністю домінують евристичні підходи. Потрібно з'ясувати чому краще використовувати систему управління контентом для сайту, замість того, щоб просто щораз створювати та модифікувати його гіпертекстовий сирцевий код. І чому веб-програмісти надають перевагу використанню саме CMS або різноманітних фреймворків, ніж написанню сайту «з нуля».

На початку розвитку Інтернету, і зокрема WWW (World Wide Web), сайти не мали складної структури, містили загалом статичні сторінки і були написані вручну в звичайному блокноті. Всім відомо, що основна мова для написання веб-сайтів – мова розмітки гіпертексту HTML (Hypertext Markup Language). При створенні сайту з великою кількістю сторінок можна було помітити, що деякі стандартні блоки HTML в структурі сайту повторюються, а тому програмістам доводилося або копіювати їх, або писати заново, що не дуже зручно. Для того, щоб спростити собі життя, вони почали використовувати спеціальні текстові редактори, які дозволяли швидко скомпонувати макет сторінки. Це давало можливість значно пришвидшити розробку сайтів.

На цьому етапі розвитку всіх все влаштовувало, можна було створювати сайти в спеціальних HTML-редакторах, однак WEB все більше розвивався, з'являлися нові технології, сайти ставали все більшими, і процес створення сайту був вже не таким швидким. Програмістам доводилося багато речей тримати в голові, де і, що знаходиться, де додати посилання, а де прибрати.

Постановка наукової проблеми Метою даної статті є аналіз найпоширеніших систем керування вмістом, та вибір оптимальної системи для подальшої роботи. Типову систему управління контентом можна розглядати як певний набір модулів, кожен із яких виконує певні функції. Контент веб-сайту формується в результаті роботи окремих модулів, а адміністрування по суті зводиться до підключення або відключення тих чи інших модулів і завдань параметрів візуального оформлення сторінки.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Чим більше WEB розвивався, тим більше він вимагав від розробників і користувачів. Почали з'являтися такі компанії, на сайтах яких треба було дуже часто і швидко оновлювати інформацію. Одній людині, та й ще в ручну було неможливо справитися з такою задачею.

Тоді, розробники почали думати як спростити цей процес, як його автоматизувати. Почали розробляти перші системи керування вмістом, для того, щоб розробникам не доводилося пам'ятати, де і що знаходиться, а змінювати всю інформацію через веб-інтерфейс.

Ця система почала буквально «думати» за веб-розробників, вона відстежувала весь вміст сайту, знала де і що зберігається, де що потрібно змінити при додаванні нової сторінки на сайт, в загальному система повністю контролювала сайт.

З часом веб-розробники почали створювати CMS для виконання різного спектру задач, від написання блогів, до створення інтернет-магазинів. Вони бувають безкоштовні і платні, а також з закритим і відкритим програмним кодом.

Системи керування контентом дозволяють значно полегшити розробку і супровід сайту. На сьогоднішній день є безліч таких систем, завдяки яким, кожен веб-програміст може підібрати найкращу саме для нього, яка буде якісно виконувати саме його задачі.

Система керування контентом WordPress.

WordPress [1] на сьогоднішній день є однією з найпопулярніших CMS. WordPress – хороша платформа для легкого створення і подальшого керування сайтом. На офіційному сайті говорять: «WordPress – це те, що ви використовуєте, коли хочете працювати над сайтом, а не боротися з ним.». Ця думка є звичайно суб'єктивною, однак WP дійсно є такою CMS, де дуже швидко, і дуже легко можна налаштувати свій проект. Буквально, зробивши пару кліків можна отримати повноцінний сайт. Тому, цю систему безумовно можна рекомендувати людям, які тільки знайомляться із системами керування контентом. Також, дана система дуже швидко встановлюється на сервер. У світі її називають «Славнозвісна установка за 5 хвилин» (майстер встановлення зображений на рис. 1).

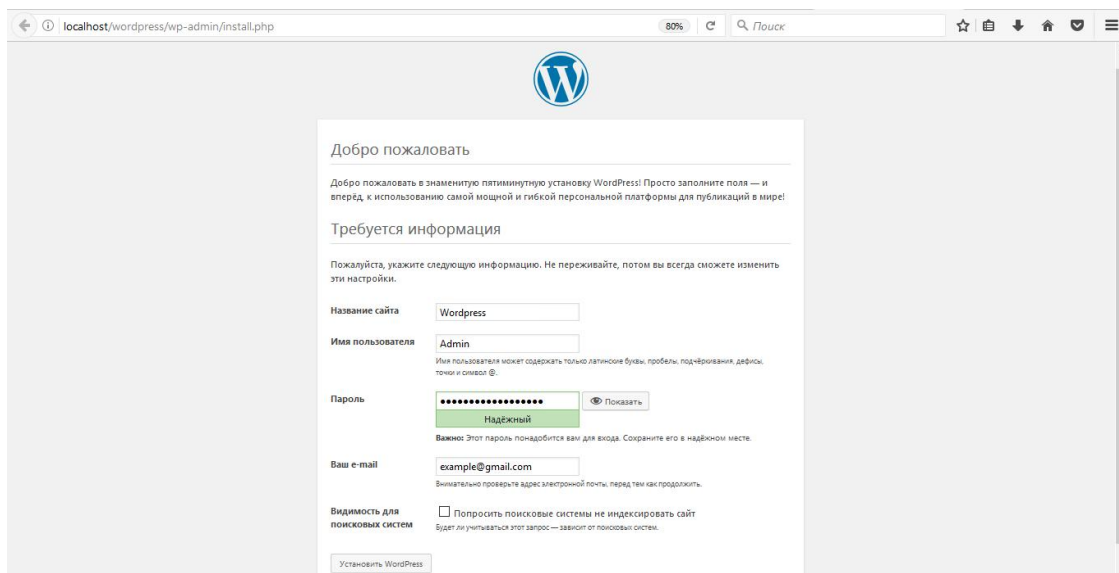


Рис. 1. Майстер встановлення WordPress

WP дає великий ресурс модулів і тем для створення сайту. Також при потребі і при певних знаннях можна написати свій модуль (плагін) або тему.

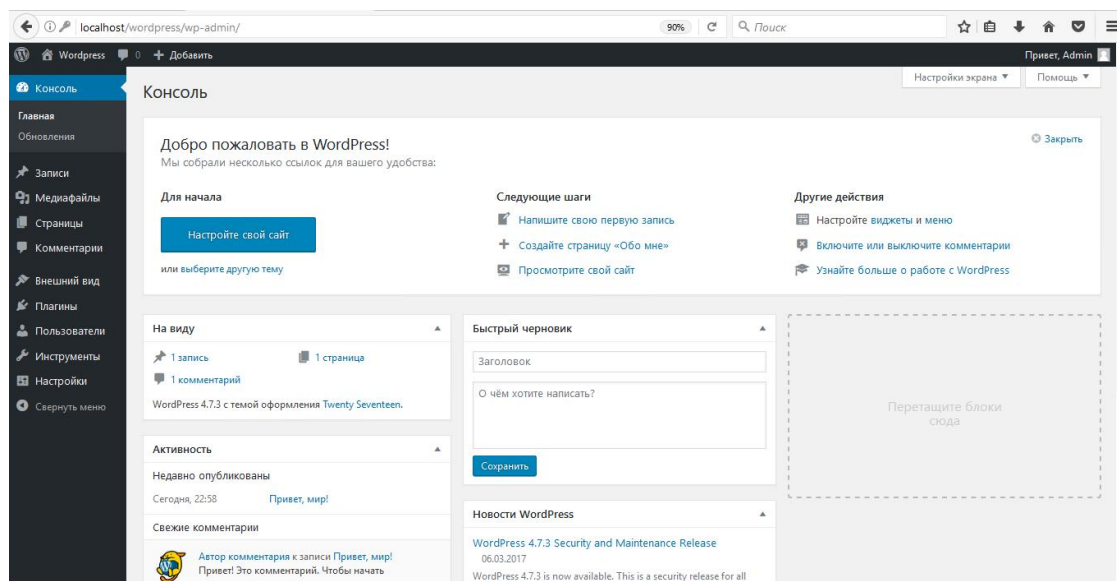


Рис. 2. Адмін-панель сайту

Вважаємо, що особливістю WP є те, що на базі цієї CMS можна побудувати веб-сайт будь-якої складності, починаючи від сайту візитки, закінчуючи потужним інтернет-магазином. Перевагою даної CMS можна вважати можливість кешування сторінок, а також підтримка SSL-протоколу, що робить сайт більш захищеним. Також перевагою саме цієї CMS можна вважати доволі велике ком'юніті.

Адмін-панель має доволі простий інтерфейс, тому навіть новачок зможе легко розібратися в цій системі. Також можна відмітити підтримку «багатомовності» сайтів, що є плюсом відносно цієї системи.

Установка тем і плагінів дуже проста, варто натиснути на відповідному пункті меню в адмін-панелі.

Система керування контентом Joomla.

Joomla є безкоштовною системою для створення веб-сайтів. Це проект із відкритим вихідним кодом, який, як і більшість подібних проектів, не стоїть на місці. Він дуже успішно розвивається, протягом останніх семи років, і користується популярністю у мільйонів користувачів у всьому світу.

На цій CMS також можна створювати різноманітні веб-сайти, різної складності, від сайтів-візиток, до сайтів галерей, або відомчих сайтів. Joomla працює з базою даних SQL. Розгорнувши цю систему на сервері, ви вже будете мати повноцінний веб-сайт, а якщо ви хочете розширити можливості сайту, то наявна можливість встановлення різних плагінів.

Встановлення системи впроваджується шляхом її розгортання на сервері, після чого в адресній стрічці прописується адреса сайту і за допомогою майстра він встановлюється. Подібні маніпуляції проводяться для встановлення сайту на CMS WordPress.

В CMS Joomla [3] є підтримка SSL, а також існує модуль безпеки для багаторівневої автентифікації, що робить цю систему доволі безпечною. Перевагою цієї системи є те, що можна обмежувати доступ до певних частин сайту незареєстрованим користувачам, чим збільшується безпечність і якість веб-сайту.

Joomla має багатомовний інтерфейс, і хорошою ознакою цієї системи також є те, що мовні пакети вмонтовані в основний дистрибутив, і користувачу потрібно тільки вибрати певну мову.

Рис. 3. Майстер встановлення сайту на Joomla

Цікавою особливістю CMS Joomla є можливість відключення сайту при проектуванні.
Система керування контентом Drupal.

У Drupal [6] є дві ролі: це система керування контентом (CMS) та так званий каркас програмної системи (FMS). В другому варіанті Drupal використовується як фреймворк, для створення веб-додатків.

Дана система підтримує бази даних MySQL та PostgreSQL. Архітектура Drupal також як і розглянутих вище CMS дозволяє створювати різноманітні сайти від блогів до соціальних мереж.

До плюсів цієї системи можна віднести:

- авторизація через OpenID;
- динамічна побудова меню;
- рольова модель (відокремлення доступу користувачам до різних частин сайту);
- мультисайтинг;
- переклад інтерфейсу сайту.

В Drupal пропонується гнучка схема організації структури сайту на основі таксономії. Таксономія – механізм, що дозволяє створювати будь-яку кількість тематичних категорій для вмісту сайту і асоціювати їх з модулями, що забезпечують введення і виведення інформації.

Механізм розширення функціоналу в Drupal [7] такий самий, як і в попередніх системах, за допомогою модулів, зовнішній вигляд міняється за допомогою тем оформлення.

З недоліків можна виділити те, що в даній системі погано реалізовано потенціал PHP, API не використовує його на повну. Також до недоліків відносять відсутність зворотної сумісності з плагінами.

В даній CMS є можливість збирати веб-статистику, є захист від автоматичного заповнення форм, підтримка протоколу.

Ця система є доволі потужною, на сьогоднішній день велика кількість людей використовує цю CMS і повністю задоволена нею. А якщо немає тої чи іншої функції, можна написати свій модуль і підключити його до системи.

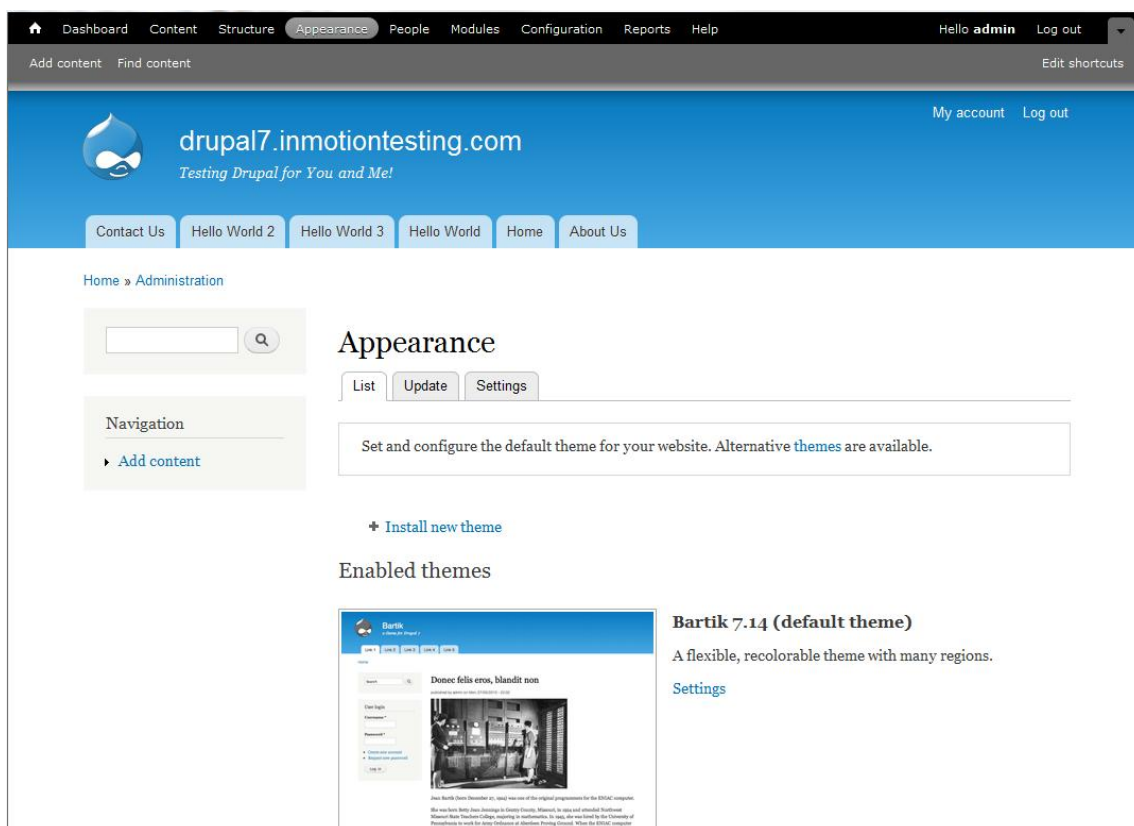


Рис. 4. Адмін-панель Drupal

Система керування контентом MODX.

Система керування вмістом MODX – це безкоштовна професійна система управління вмістом (CMS) і фреймворк для веб-додатків, призначена для забезпечення і організації спільного процесу створення, редагування і управління контентом (тобто вмістом) сайтів.

MODX поширюється безкоштовно за ліцензією GPL з відкритим вихідним програмним кодом (Open Source). Це означає, що систему MODX може використовувати кожен: як для особистого вжитку, так і для комерційного поширення сайтів, побудованих на даній системі управління.

Дана система використовується для створення різних категорій сайтів.

Деякі вбудовані функції даної CMS:

- система реєстрації користувачів;
- WYSIWYG – редактор;
- Підтримка модульної розробки;
- Система публікації коментарів (з модерацією і підпискою);
- Генерація каталогів, блогів, новин.

До переваг даної CMS можна віднести:

- графічний веб-установник;
- легко розширити можливості сайту за допомогою плагінів або сніппетів;
- сумісність з MVC;
- можливість розгорнути CMS в хмарі;
- кросбраузерність і кросплатформеність роботи;
- репозиторій готових розширень;
- магазин додатків для MODX

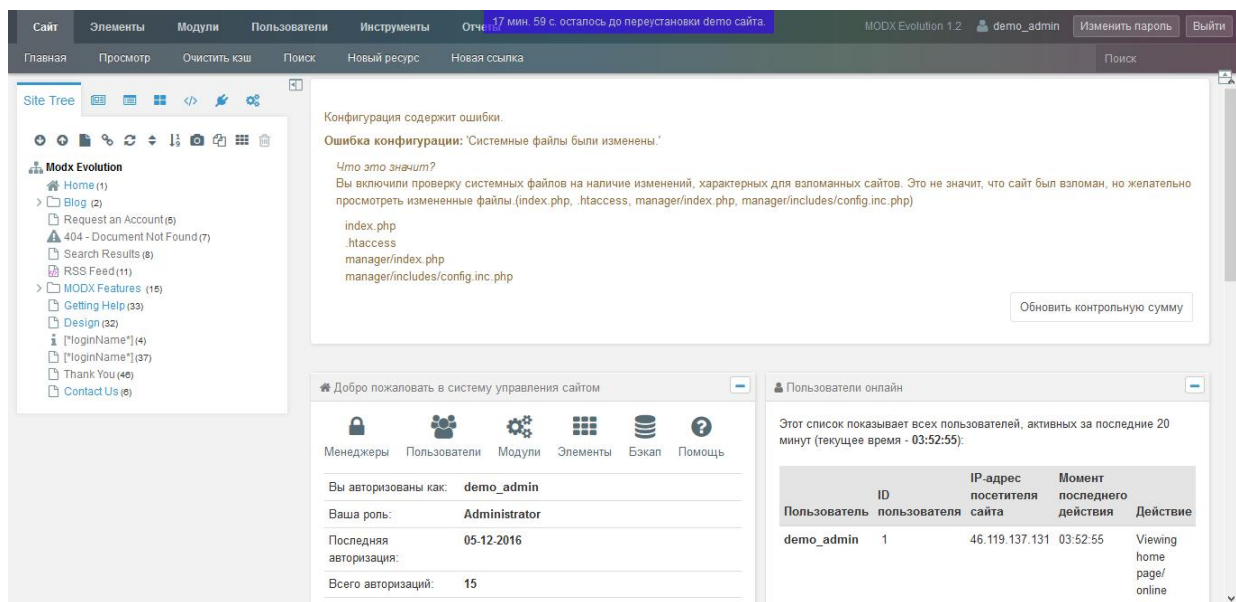


Рис. 5. Адмін-панель MODX

Вважаємо, що дана CMS є однією з найпотужніших систем керування вмістом. Однак, єдиним і найбільшим її мінусом є те, що, вона не розрахована на новачків. Користувачу, який не знається на програмуванні буде важко управляти цією CMS.

Таблиця 1 – Результати аналізу

Критерії	WordPress	Joomla	Drupal	MODX
Ведення запису дій користувача	-	-	+	+
Захист від автоматичного заповнення фор	?	+	+	+
Кешування сторінок	+	+	+	+
Розподіл прав користувачів	-	+	+	+
Підтримка SSL-протоколу	+	+	+	+
Декілька мов інтерфейсу	+	+	+	+
Підтримка «багатомовності» сайтів	+	+	+	+
Веб-статистика	+	+	+	+
Чат	-	+	+	?
Форум	-	+	+	-
Фотогалерея	+	+	+	+
Каталог продукції	-	+	+	+
Платіжні системи	+	+	+	+
Легкість керування	+	+	-	-

Висновки.

Обґрунтовано, що найкращим вибором для керування сайтом є CMS Joomla. Ця система має зручний інтерфейс, легко встановлюється і в підсумку аналізу має переваги за всіма

© Пастернак Я.М., Братанов А.С.

критеріями відбору. Крім того, її можна використати для різних категорій сайтів – від блогів до інтернет-магазинів.

На другому місці після Joomla вважаємо доцільним розмістити CMS MODX, яка є доволі потужною професійною системою, що дуже пригодиться професійним розробникам веб-сайтів та веб-додатків.

1. Scott, Adam D. WordPress for Education. — Birmingham: Packt Publishing Ltd, 2012. — 144 с. — ISBN 978-1-84951-820-8.
2. Antano Solar John. MODx Web Development. — Second Edition. — PACKT, February 23, 2011. — 288 p. — ISBN 978-1849513487. (—//—. — 1st edition. — March 2009. — 276 p. — ISBN 978-1-847-19490-9.)
3. Дэн Рамел. Joomla! для профессионалов = Advanced Joomla!. — М.: «Вильямс», 2014. — 448 с. — ISBN 978-5-8459-1891-8.
4. Колисниченко Денис Николаевич. Joomla! 3.0. Руководство пользователя. — М.: «Диалектика», 2013. — 256 с. — ISBN 978-5-8459-1864-2.
5. Марк Декстер, Луис Лэндри. Joomla!: программирование = Joomla! Programming (Joomla! Press). — М.: «Вильямс», 2013. — 592 с. — ISBN 978-5-8459-1798-0.
6. Тодд Томлинсон. CMS Drupal 7: руководство по разработке системы управления веб-сайтом, 3-е издание = Pro Drupal 7 Development, 3rd Edition. — М.: «Вильямс», 2011. — 560 с. — ISBN 978-5-8459-1743-0.
7. Мелансон Б., Нордин Д., Луиси Ж. Профессиональная разработка сайтов на Drupal 7. — СПб.: «Питер», 2013. — 688 с. — ISBN 978-5-4461-0054-5.
8. Алексей Черных. Drupal 7. — «Эксмо», 2011. — С. 208. — ISBN 978-5-699-47059-4 rus.
9. Колисниченко Денис Николаевич. Drupal 7. Руководство пользователя. — «Диалектика», 2011. — С. 256. — ISBN 978-5-8459-1717-1.
10. Ромашов В. CMS Drupal: система управления содержимым сайта (+CD с видеокурсом). — СПб.: «Питер», 2010. — С. 256. — ISBN 978-5-49807-241-8.

УДК 004.31

Пех П. А., Баландюх В. В.

Луцький національний технічний університет

РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ КОСМЕТИКИ ТА ЗАСОБІВ ГІГІЄНИ НА ОСНОВІ CMS WORDPRESS

Пех П. А., Баландюх В. В. Розробка інтернет-магазину з продажу косметики та засобів гігієни на основі CMS WordPress. Розроблено інтернет-магазин з продажу косметики та засобів гігієни на основі CMS WordPress та з використанням плагіна Woocommerce.

Ключові слова: інтернет-магазин, WordPress, Woocommerce, браузер, сервер, хост, eCommerce, URL-адрес, PHP, CSS, HTML.

Пех П. А., Баландюх В. В. Разработка интернет-магазина по продаже косметики и средств гигиены на основе CMS WordPress. Разработан интернет-магазин по продаже косметики и средств гигиены на основе CMS WordPress и с использованием плагина Woocommerce.

Ключевые слова: интернет-магазин, WordPress, Woocommerce, браузер, сервер, хост, eCommerce, URL-адрес, PHP, CSS, HTML.

Peh, P. A., Balandiukh V. V. Development of online store selling cosmetics and personal care products based on CMS Word Press. Is developed online store selling cosmetics and personal care products based on CMS WordPress using the Woocommerce plugin.

Keywords: online store, WordPress, Woocommerce, browser, server, host, eCommerce, URL-address, PHP, CSS, HTML.

Вступ. У 1990 році Тім Бернерс-Лі створив перший веб-сервер і браузер. Він був відкритий для комерційного використання в 1991 році. У 1994 році відбулися інші досягнення, 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.

Важливі елементи інтернет-магазину — оновлення наявного асортименту (продукти/товари та їх кількість), можливість додавати товари до «кошику», вхід для зареєстрованих користувачів. У деяких випадках можна використовувати систему оплати через інтернет (електронні гроші), у простішому випадку оплата відбувається звичайними грошми через банк на рахунок, роздрукований з сайту. Оплата через інтернет передбачає необхідність створення кількох облікових записів (принаймні двох), але у деяких випадках системи оплати можуть бути пов'язаними з магазином.

Для того, щоб інтернет-магазин видавав інформацію залежно від запитів, надавав можливості пошуку тощо — на сервер встановлюється підтримка скриптів (наприклад PHP, Perl). У більш комплексному варіанті, програма інтернет-магазину — це система управління вмістом сайту, яка вже має підтримку скриптів, надає можливість в он-лайн режимі (головним чином через інтернет) і в межах наявного асортименту виконувати купівлю потрібних товарів. Саме розробці всіх цих пунктів присвячена дана робота.

Постановка наукової проблеми. Метою даної роботи є аналіз існуючих методів створення інтернет-магазинів і вибір за певними критеріями найбільш ефективних з них у процесі розроблення сайту відповідного напрямку.

Актуальність. Розробка інтернет-магазину з продажу косметики та засобів гігієни на CMS Word Press є актуальною, оскільки на даний час все більше і більше людей користуються інтернет-магазинами, оскільки це дуже зручно, бюджетно та ефективно як і власнику інтернет-магазину, так і покупцю. Інтернет-магазин не потребує значних витрат на оренду приміщення та залучення на роботу працівників, а тим більше не потребує значних комунальних витрат.

Формулювання цілі. Розробити інтернет-магазин з продажу косметики та засобів гігієни на CMS Word Press.

Аналіз існуючих способів.

Magento — система, яка розрахована на великі магазини, для підтримки яких потрібна багатофункціональність. *Magento* є найбільш універсальною системою з тих, які ми сьогодні

будемо розглядати, тому що вона дозволяє виконувати велику кількість операцій eCommerce з великою кількістю налаштувань, необхідних для цього.

Вона може обробляти тисячі продуктів з використанням різних валют, платіжних систем, різних мов і вітрин магазинів – вам доступні будь-які функціональні можливості для досягнення цього. Єдиний мінус даної системи це велика ціна.

Система Magento розрахована на великі магазини, для підтримки яких потрібна багатофункціональність. Magento є найбільш універсальною системою, тому що вона дозволяє виконувати велику кількість операцій eCommerce з великою кількістю налаштувань, необхідних для цього.

Незважаючи на те, що back-end системи має відмінний дизайн, тим, хто не знайомий з подібною функціональністю, доведеться витратити чимало часу на прочитання документації кілька разів і поламати голову над налаштуванням сайту. Magento явно не годиться для новачків...

Вартість розширень також не порадує. Якщо ви звикли до безкоштовних тем або плагінів WordPress за \$19, то тоді вам буде важко змиритися з тим, щоб віддати кілька сотень доларів. Якщо вам потрібні користувацькі функції, створені програмістом, то за них доведеться заплатити набагато більше, ніж на WordPress, тому що система Magento більш масштабна, і з нею не так легко працювати.

Ще один момент – це витрати на хостинг. Система буде працювати повільно на звичайному віртуальному хостингу (shared hosting), тому вам знадобиться виділений сервер (dedicated server) або гарні налаштування хмарного хостингу.

OpenCart – якщо ви шукаєте більш просте рішення eCommerce, то альтернативним варіантом може бути OpenCart. Ця система управління контентом набагато простіше в установці і роботі. Разом з цим вона містить достатню кількість можливостей, тим і доповнень для магазинів у сфері eCommerce. На даний момент сайт OpenCart містить більше 14 000 розширень, включаючи понад 2700 тем – від безкоштовних до доступних за ціною, що загалом набагато дешевше ніж Magento.

Для OpenCart існує величезна кількість додатків, і багато з них – безкоштовні. Ви зможете налаштувати і працювати з сайтом, використовуючи різні платіжні системи, не витративши ні копійки. А на інших платформах вам потрібно за них заплатити.

Як вже було сказано, чим менша вартість тим і розширень менше, і вибір безкоштовних доповнень досить великий. Швидше за все, це завдяки тому, що система не дуже складна для вивчення. Консоль системи є зручною, і вона набагато зрозуміліша, ніж система Magento. У OpenCart є непогана документація разом з навчальними відеоматеріалами.

Ця система може працювати повільно, навіть на виділеному сервері. Але все можна виправити, якщо використовувати плагін кешування. З його допомогою можна зазначити окремі частини сайту для збереження у вигляді HTML, а інші за допомогою PHP.

Також можуть виникнути труднощі при внесенні змін до використовуваної вами теми. Структура шаблону досить складна для розуміння, а частини коду знаходяться в базі даних, а не в HTML, як в темах WordPress. Тому будьте готові до того, що вам потрібно буде чимало попрацювати, щоб зробити основні зміни у використовуваної вами темі.

Ще один нюанс у тому, що сценарій системи не виробляє пошук дружніх URL-адрес, так що знову ж таки, вам буде потрібне розширення, щоб виправити це.

WordPress + Woocommerce – система була запущена WooThemes в 2011 році. Вона має величезну кількість плагінів і тем, а також додаткових бонусів, оскільки побудована на платформі WordPress. Більшість плагінів можуть бути також використані з нею, що значно розширює кількість додатків для кінцевого користувача.

Переваги WooCommerce і WordPress, як системи управління контентом в цілому. Це: екосистема, велика кількість розробників, безліч безкоштовних тем і плагінів, дуже простих в установці і використанні, постійні оновлення, використання low-end хостингу, оптимізація під пошукові системи, що є необхідним при використанні WooCommerce для онлайн-магазину.

Якщо говорити виключно про WooCommerce, то ця система дуже проста в установці – ви встановлюєте WordPress і активуєте плагін, який створює необхідні вам сторінки. У цій системі є дуже зрозумілий розділ з документацією на сайті і величезна кількість доступних розширень.

WooCommerce містить не настільки багато основних функцій, скільки інших параметрів, тому вам потрібно буде придбати за додаткову плату досить багато додаткових розширень для одержання необхідної вам функціональності.

Якщо вам потрібна буде допомога підтримки, то доступ до потрібного форуму також платний. Якщо ж ви не звикли використовувати сайт на WordPress, то вам доведеться витратити час, щоб пристосуватися до нього та до системи шаблонів тим.

Мінуси використання WooCommerce також відносяться і до WordPress – з-за популярності платформи, яка провокує спроби хакерів зламати систему, вам потрібно буде стежити за оновленнями і встановлювати додаткові рівні безпеки для свого сайту. Також необхідно стежити, щоб оновлення коду не впливало негативно на функціональність плагіна.

Враховуючи все вищесказане і, виходячи з практичного досвіду користувачів, можна сказати, що системи Magento і OpenCart працюють досить легко. Але вивчення документації і пошук потрібних розширень забирає багато часу.

Якщо ж ви збираєтеся вести великий сайт в сфері eCommerce, то придивіться більше до системи Magento, оскільки вона дійсно потужна і дозволяє підтримувати великі онлайн-магазини з допомогою функціоналу, який не забезпечить більше жодна система. Але в цьому випадку будьте готові витратитися на доповнення.

Виклад основного матеріалу:

Перш ніж побачити інтернет-магазин в всесвітній павутині, для початку потрібно висвітити структуру сайту. Структурою сайту слугує створення його прототипу на папері чи будь якому підручному засобі, яким ви привикли користуватися. В даному випадку креслення відбувається у програмі Paint (див. схема 1.1).

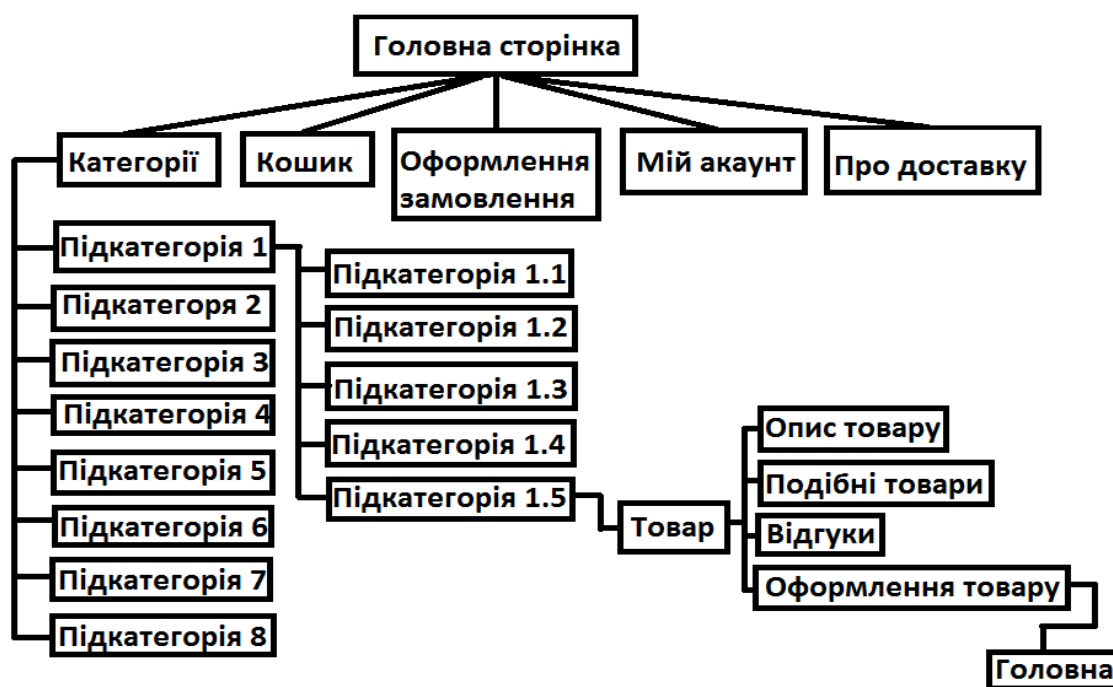


Схема. 1.1 Структура сайту

Наступним кроком буде створення теми сайту. Структура сайту написана на HTML-мові програмування. Дизайн сайту оформлений за допомогою CSS – мові програмування.

Головна сторінка сайту компанії - це обличчя сайту, звернене до всієї мережі. Саме з головної сторінки більшість користувачів починають свою роботу зі сайтом. Покращуючи дизайн головної сторінки ви підвищуєте всю цінність сайту. Головна сторінка складається з таких частин: header, main, sidebar, footer (див. схема 1.2).

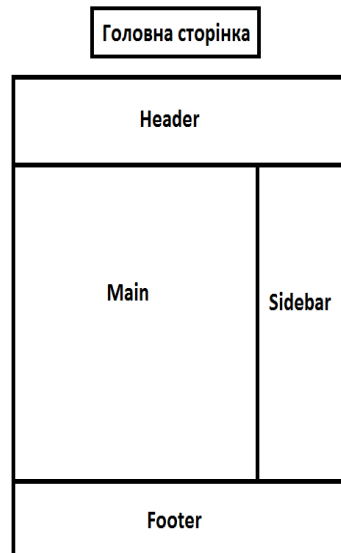


Схема 1.2 Структура головної сторінки.

Header – голова сайту, зазвичай в даному розділі розміщується логотип компанії та спосіб зв'язку з нею (мал.1.1).

```

<header id="masthead" class="site-header" role="banner">
  <div class="hgroup full-container">
    ::before
    <a href="http://kosmasha.com.ua/" title="" kosmasha – Інтернет-магазин косметики" rel="home"
      class="logo">
       == $0
    </a>
    <div class="support-text">
      Зателефонуйте нам просто зараз - 095-043-6669    </div>
    ::after
  </div>
  <!-- .hgroup.full-container -->
  <nav role="navigation" class="site-navigation main-navigation primary use-sticky-menu">
    <div class="full-container">
      <div id="search-icon">
        <div id="search-icon-icon"></div>
        <form method="get" class="searchform" action="http://kosmasha.com.ua/" role="search"
          style="width: 1080px;"></form>
      </div>
      <div class="menu-%d0%b3%d0%be%d0%bb%d0%be%d0%b2%d0%bd%d0%b5-%d0%bc%d0%b5%d0%bd%d1%8e-
        container"></div>
      </div>
    </nav>
    <!-- .site-navigation .main-navigation -->
  </header>
  <!-- masthead .site-header -->
  
```

Мал. 1.1 Header сайту.

Main – основна частина, в ній розміщується весь каталог товарів.

Sidebar – бічна частина сайту, в ній розміщуються різні віджети, в даному випадку це фільтр по ціні, спільнота в соціальній мережі та при додаванні в кошик товару, він відображається в даній частині (Мал.1.2).

```

<aside id="woocommerce_price_filter-2" class="widget woocommerce widget_price_filter"> ==
  <h3 class="widget-title">Фільтр по ціні</h3> $0
  <form method="get" action="http://kosmasha.com.ua/">
    <div class="price_slider_wrapper">
      <div class="price_slider ui-slider ui-slider-horizontal ui-widget ui-widget-content
        ui-corner-all" style">
        <div class="ui-slider-range ui-widget-header ui-corner-all" style="left: 0%;
          width: 100%;"></div>
        <span class="ui-slider-handle ui-state-default ui-corner-all" tabindex="0" style="
          left: 0%;"></span>
        <span class="ui-slider-handle ui-state-default ui-corner-all" tabindex="0" style="
          left: 100%;"></span>
      </div>
      <div class="price_slider_amount">
        <input type="text" id="min_price" name="min_price" value data-min="15"
          placeholder="Мінімальна ціна" style="display: none;">
        <input type="text" id="max_price" name="max_price" value data-max="300"
          placeholder="Максимальна ціна" style="display: none;">
        <button type="submit" class="button">Фільтрувати</button>
        <div class="price_label" style=""></div>
        <div class="clear"></div>
      </div>
    </div>
  </form>
</aside>
  
```

Мал. 1.2 Sidebar сайту.

Footer – нижня частина, в ній розміщується повна контактна інформація компанії, які послуги надає дана компанія, та способи оплати (Мал.1.3.)

```
<?php if ( !isset( $colophon ) ) :  
    <div id="colophon" class="site-footer" role="contentinfo"> == $0  
    <div id="footer-widgets" class="full-container">  
        <div id="text-3" class="widget widget_text">  
            <div class="textwidget">  
                <p>  
                    "Інтернет-магазин з продажу косметики і засобів гігієни"  
                    <br>  
                    "Контакти: e-mail: kosmasha1997@gmail.com"  
                    <br>  
                    "Телефон: 095-043-6669"  
                </p>  
            </div>  
        </div>  
    </div>  
    <div id="theme-attribution"> </div>  
</div>  
<!-- #colophon .site-footer -->  
</div>  
<!-- #page-wrapper -->
```

Мал. 1.3 Footer сайту.

Висновки

Проаналізувавши способи розробки інтернет-магазину, дійшли висновку, що найбільш оптимальний і бюджетний спосіб – це інтернет-магазин з використанням WordPress і Woocommerce. Інші способи нам не підійшли Magento – дорогий у використанні, а OpenCart – повільний у роботі.

Що ж стосується WooCommerce, то з ним все набагато простіше, особливо якщо у нас є потрібний WordPress плагін екосистеми для додавання функціональності на сайт. Можна придбати ще кілька плагінів для додаткових функцій, але в цьому немає особливої необхідності, оскільки оновлення платних плагінів може негативно вплинути на оновлення WordPress в майбутньому.

1. <https://uk.wikipedia.org/wiki/WordPress>
2. <https://wpcafe.org/tutorials/wordpress-magento-opencart-woocommerce/>
3. <http://kosmasha.com.ua/>
4. <https://ru.wordpress.org/plugins/woocommerce/>

УДК 65.011.46

Ходаков В.Е. д.т.н., профессор, Козел В.Н. старший преподаватель
Херсонский национальный технический университет

ОЦЕНКА ЭФФЕКТИВНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ ПО ЧАСТНЫМ ПОКАЗАТЕЛЯМ КАЧЕСТВА

Ходаков В.Е., Козел В.Н. Оценка эффективности информационных систем по частным показателям качества. В статье рассмотрены основные частные показатели оказывающие непосредственное влияние на информационные системы. Даны определения адекватности, релевантности информации, толерантности. Представлен комплексный показатель эффективности ИС.

Ключевые слова: частные показатели качества, эффективность системы, информационная система, адекватность, критерии оценки информации.

Ходаков В.Є., Козел В.М. Оцінка ефективності інформаційних систем по приватним показниками якості. У статті розглянуті основні приватні показники які безпосередньо впливають на інформаційні системи. Дано визначення адекватності, релевантності інформації, толерантності. Представлений комплексний показник ефективності ІС.

Ключові слова: приватні показники якості, ефективність системи, інформаційна система, адекватність, критерії оцінки інформації.

Khodakov V.E., Kozel V.M. Evaluation of the effectiveness of information systems on private quality indicators. The article deals with the main private indicators that have a direct impact on information systems. The definitions of adequacy, relevance of information, tolerance are given. A complex index of efficiency is presented.

Key words: private quality indicators, system efficiency, information system, adequacy, criteria for information evaluation.

Введение. Определение эффективности информационных системы является сложным и трудоемким процессом.

При современном развитии технических средств для решения одной и той же задачи допускается несколько различных вариантов построения и функционирования ИС, каждый из которых реально осуществим. Поэтому задача оценки эффективности ИС сводится к поиску оптимальной системы [1,2]. Окончательное решение можно принять только на основании расчетов по оптимизации или опытных проверок, сводящихся к отысканию варианта с экстремальными или относительно лучшими значениями критерия, соответствующее искомому оптимуму.

Постановка задачи. Сравнение различных критериев эффективности приводит к выводу, что наиболее приемлемыми являются критерии, позволяющие при знании весовых коэффициентов частных показателей качества системы определить практически оптимальную систему. Под практически оптимальной системой [1] понимается система, которая имеет наивысший показатель эффективности среди рассматриваемых систем, даже если ни при одном из частных показателей качества (ЧПК) не достигнуто экстремальное значение. Значение весовых коэффициентов переводит задачу оптимизации в разряд алгебраических, что значительно упрощает ее решение.

Целью статьи является определение оценки эффективности сложных информационных систем с использованием весовых коэффициентов частных показателей качества.

Основной материал. В большинстве случаев наиболее удобным является способ оценки систем по обобщенному показателю эффективности, представляющему собой линейную функцию от частных показателей качества. По такому обобщенному показателю сравнительно просто оценить конкурирующие варианты систем и выбрать практически оптимальный вариант. Основная трудоемкость при использовании данного способа заключается в определении численных значений весовых коэффициентов, с которыми частные показатели качества входят в обобщенный показатель эффективности системы.

Определение эффективности системы по показателю эффективности, заданному аналитически.

В общем случае эффективность внедрения или модернизации системы оценивается показателем

$$E = D/C, \quad (1)$$

где D – эффект, величина, показывающая, что дает применение новой системы или модернизация существующей, C – затраты на разработку, внедрения и эксплуатацию или модернизацию системы.

Теоретически показатель эффективности новой информационной системы E учитывают все затраты, однако не все эти показатели могут быть непосредственно измерены. В качестве одного из методов определения эффективности ИС предлагается заменять функцию показателя эффективности ИС, если неизвестно ее аналитическое значение, линейной функцией, включающей все основные частные показатели качества ИС.

$$E = b_1 y_1 + b_2 y_2 + \dots + b_n y_n, \quad (2)$$

где $b_1, b_2, \dots, b_n$ – весовые коэффициенты. $y_1, y_2, \dots, y_n$ – частные показатели качества.

Отдельной задачей при использовании линейной функции (2) является определения перечня учитываемых (основных) частных показателей качества системы y_i и их весовых коэффициентов b_i . Определение этих показателей трудоемкая задача, которая практически сводит на нет решение задачи по оценки эффективности.

Оценка эффективности по частным показателям качества.

Для удобства сравнения нескольких вариантов модернизации по различным критериям их количественные показатели выражаются в относительных единицах, при этом сравнивается абсолютное значение показателя качества модернизированной системы с абсолютным значением этого же показателя, оговоренного техническими требованиями к системе, или, если эти требования не оговорены, с показателем исходной модернизируемой системы [4]

В качестве частных показателей качества системы применяют их относительные показатели

$$\bar{y}_i = \frac{y_{i\text{ факт}}}{y_{i\text{ доп}}}, \quad (3)$$

где $y_{i\text{ доп}}$ – допустимое значение i -го показателя качества исходной системы; y_i – абсолютное значение этого показателя для модернизированной системы.

Используя данный метод расчета эффективности сложных систем позволяет привести частные показатели качества к абсолютному значению, что уравнивает все критерии.

Оптимизация систем по одному показателю иногда бывает достаточной, и может быть рассчитана аналитическими или графическими методами. Однако эффективность, определенная таким образом, дает одностороннюю характеристику системы, так как не учитывает всей совокупности факторов, влияющих на выбор системы. Если рассматриваемая система является частью некоторой большой системы, то любое ее улучшение будет оправданным только в том случае, если при этом повышается эффективность всей большой системы [2].

Комплексный показатель эффективности информационных систем E_k представляет собой линейную комбинацию единичных критериев E_i

$$E_k = \sum_{i=1}^n b_i E_i, \quad (4)$$

где n – число учитываемых единичных критериев, b_i – коэффициент.

Комплексный показатель эффективности ИС не учитывает назначение информационной системы и поэтому не может служить для уверенного выбора лучшего варианта самонастройки, так как самонастраивающаяся система сама является частью какой-то большой системы. Поэтому получение оптимальных результатов по самонастройке может не означать получение оптимальных результатов по всему комплексу.

Для определения частных показателей качества, любой информационной системы необходимо рассмотреть основные показатели оценки информации как ресурса. Одними из таких показателей качества является «важность», «полнота информации», «адекватность информации».

Важность информации должна оцениваться по двум группам критериев (рис. 1):

- по назначению информации;
- по условиям ее обработки.

В первой группе следует выделить два критерия:

- важность самих задач для обеспечения деятельности,
- степень важности информации для эффективного решения соответствующей задачи.

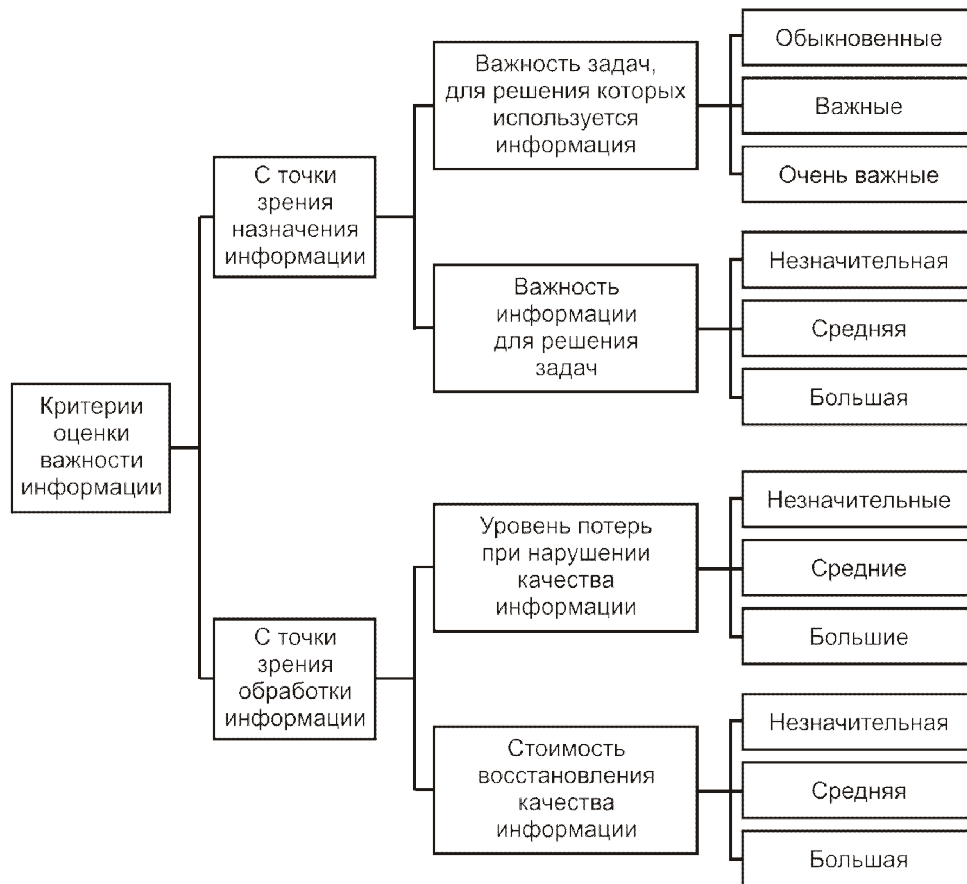


Рис. 1. Критерии оценки важности информации

Во второй группе выделяются два составных критерия:

- уровень потерь в случае нежелательных изменений информации в процессе обработки под воздействием дестабилизирующих факторов,
- уровень затрат на восстановление нарушенной информации.

Если обозначить: $K_{ви}$ — коэффициент важности информации; $K_{вз}$ — коэффициент важности тех задач, для обеспечения которых используется информация; $K_{из}$ — коэффициент важности оцениваемой информации для эффективного решения задач; $K_{пи}$ — коэффициент важности оцениваемой информации с точки зрения потерь при нарушении ее качества; $K_{св}$ — коэффициент важности информации с точки зрения стоимости восстановления ее качества. Тогда получим[1]:

$$K_{ви} = f(K_{вз}, K_{из}, K_{пи}, K_{св}) \quad , \quad (5)$$

Иначе говоря, для оценки важности информации необходимо уметь определять значения перечисленных выше коэффициентов и знать вид функциональной зависимости $K_{ви}$. Но на сегодняшний день неизвестно ни то, ни другое. Однако иногда для этих целей для конкретной информации и конкретных условий можно использовать подход, основанный на неформально-эвристических методах.

Полнота информации — это показатель, характеризующий меру достаточности оцениваемой информации для решения соответствующих задач. Так же, как и предыдущий показатель, является относительным: полнота информации оценивается относительно вполне определенной задачи или группы задач. Поэтому чтобы иметь возможность определить показатель полноты информации, необходимо для каждой существенно значимой задачи или группы задач составить перечень тех сведений, которые необходимы для их решения.

Под *адекватностью информации* понимается степень ее соответствия действительному состоянию тех объектов, процессов или явлений, которые отображает оцениваемая информация. В общем случае адекватность информации определяется двумя параметрами.

1. Объективностью генерирования (съема, определения, установления) информации об объекте, процессе или явлении.

2. Продолжительностью интервала времени между моментом генерирования информации и моментом оценки ее адекватности.

Объективность генерирования информации, очевидно, зависит от способа получения значений характеристик объекта, процесса или явления и качества реализации (использования) способа в процессе получения этих значений. Классификация характеристик по возможным способам получения их значений представлена на рис. 2.

Рассмотрим теперь адекватность информации по второму названному параметру — продолжительности интервала времени между моментом генерирования информации и текущим моментом. Для оценки адекватности по данному параметру вполне подходящим является известный в теории информации так называемый закон старения информации (рис.3).

При этом под t_0 понимается момент времени генерирования (получения) оцениваемой информации; Δt_1 — продолжительность времени, в течение которого оцениваемая информация полностью сохраняет свою адекватность; Δt_2 — продолжительность времени, в течение которого адекватность информации падает на 25%; Δt_3 — продолжительность времени, в течение которого адекватность информации падает наполовину; Δt_4 — продолжительность времени, в течение которого адекватность падает на 75%. [5]

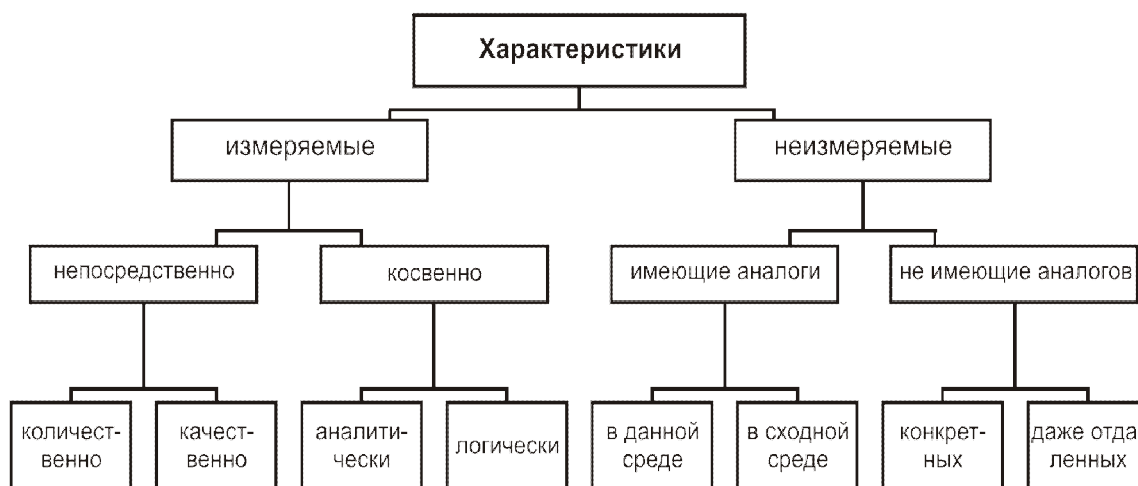


Рис.2. Классификация характеристик по способам получения их значений



Рис. 15.2. Классификация характеристик по способам получения их значений

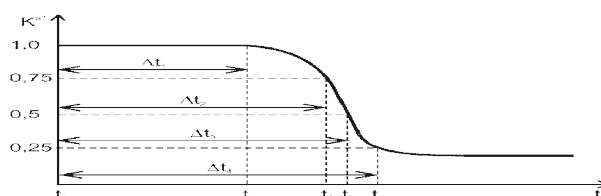


Рис.3. Графическое представление закона старения информации

Учитывая то, что обе составляющие адекватности информации K_a' и K_a'' зависят от большого числа факторов, многие из которых носят случайный характер, есть основание утверждать, что они также носят случайный характер и поэтому могут интерпретироваться как вероятности того, что информация по соответствующему параметру является адекватной.

Поскольку для подавляющего большинства теоретических исследований и практических приложений важно, чтобы информация была адекватной одновременно по обоим параметрам, то в соответствии с теоремой умножения вероятностей общий показатель адекватности информации может быть определен как: $K_a = K_{a'} \cdot K_{a''}$. Независимость значений, a' и a'' представляется вполне естественной.

Релевантность информации — это показатель, который характеризует соответствие ее потребностям решаемой задачи. Для количественного выражения данного показателя обычно используют так называемый коэффициент релевантности K_r — отношение объема релевантной информации N_r к общему объему анализируемой информации N_o , т. е. $K_r = N_r / N_o$. Сущность коэффициента релевантности очевидна, но трудности практического его использования сопряжены с количественным выражением объема информации. Поэтому задача вычисления этого коэффициента на практике относится к весьма неопределенной и трудноразрешимой проблеме.

Толерантность информации — это показатель, характеризующий удобство восприятия и использования информации в процессе решения той задачи, для решения которой она используется. Уже из самого определения видно, что понятие толерантности является очень широким, в значительной мере неопределенным и субъективным.

Перечисленные показатели информации, являются субъективными оценками качества информационных систем, и на прямую зависят от экспертов, проводящих оценку той или иной информационной системы. Однако, данный подход оценки эффективности информационных систем позволяет оценить ИС в кратчайшие сроки, опираясь на показатели.

Выводы.

Все показатели качества информации можно разделить на три категории: определяющие, существенные и второстепенные, причем основным критерием для такого деления должна служить та цель, для достижения которой предназначена информационная система.

Требуемый уровень эффективности информационной системы устанавливается по значениям определяющих показателей информации. В зависимости от поставленной задачи перед информационной системой.

При необходимости можно скорректировать оценку эффективности ИС с учетом значения существенных показателей. Значения второстепенных показателей при этом могут игнорироваться.

Решив задачу количественной оценки каждого показателя информации, позволяет свести оценку эффективности ИС к линейной комбинации единичных критериев.

1. Исаев Георгий Николаевич. Информационные системы в экономике : учебник для студентов вузов, обучающихся по специальностям «Финансы и кредит», «Бухгалт. учет, анализ и аудит» / Г. Н. Исаев. — 3-е изд., стер. — М. : Издательство «Омега-Л», — 462 с.
2. Зайченко Ю.П. Основы проектирования интеллектуальных систем: Навч. посібник. — К.: Видавничий Дім „Слово”, 2004. — 353 с.
3. Хорошко В.А., Чекатков А.А. Методы и средства защиты информации, К.: Юниор, 2003. — 476 с.
4. Планирование развития ИТ на базе методологии Balanced Scorecard //Корпоративный менеджмент. Информационные технологии в управлении, 1998-2003.
5. Згуровський М.З., Панкратова Н.Д. Основы системного аналізу -К.: Видавнича група ВНУ, 2007. — 546с.
6. Введение в нормативную теорию принятия решений : методы и модели : Монография / Виктор Владимирович Крючковский, Эдуард Георгиевич Петров, Надежда Андреевна Соколова, Виктор Егорович Ходаков; Под ред. Эдуард Георгиевич Петров.— Херсон : Гринь Д. С., 2013.— 282 с.
7. Высшее образование: взгляд со стороны и изнутри / Виктор Егорович Ходаков.— Херсон : Олди-плюс, 2001.— 214 с.
8. Волков, И.М. Критерии оценки проектов / И.М. Волков, М.В. Грачева, Д.С. Алексанов // Институт экономического развития Всемирного банка.— М.: 2006. —с. 128.
9. Бунова Елена Вячеславовна, Буслаева Ольга Станиславовна Оценка эффективности внедрения информационных систем // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. 2012. №1
10. Мартынов, О. Ю. Методика расчета эффективности от внедрения информационных технологий / О. Ю. Мартынов, Ю. В. Лохан // PDM-технологии: внедрение, эффективность: материалы науч.-практ. конф., Москва 15–17 декабря 2008 г. / Москва, гос. ун-т; редкол.:М. А. Швайцер [и др.]. — М., 2008. — С. 36–39.
11. Gupta J.N.D., Sharma S., Rashid M.A. Handbook of Research on Enterprise Systems. – Farmington Hills: Gale Group, 2009. – 460 p
12. Зайченко, О. Ю. Дослідження операцій [Текст] : збірник задач / О.Ю. Зайченко, Ю.П. Зайченко. - К. : Слово, 2007. - 472 с.

УДК 004.94

Гоян Д.І., бакалаврант, Краснокутська І.В., к.ф.-м.н.

Чернівецький національний університет імені Юрія Федьковича

СИСТЕМА АВТОМАТИЗАЦІЇ ОБЛІКУ ШВЕДСЬКО-УКРАЇНСЬКОГО МЕДИЧНОГО ЦЕНТРУ ANGELHOLM

Гоян Д.І., Краснокутська І.В. Система автоматизації обліку шведсько-українського медичного центру Angelholm. Керування та облік створюють добрий імідж установи, що надає широкий спектр медичних послуг. Дана робота присвячується розробці програмного забезпечення для клініки Angelholm з метою підвищення зручності роботи персоналу та швидкості обслуговування пацієнтів.

Ведення документації лікарем займає надто багато часу, він заповнює безліч реєстраційних форм, протоколів, карток. Запропонована нами спеціалізована система дає змогу легко вести повний медичний документообіг, облік роботи лікарів та черги пацієнтів із використанням сучасних інформаційних технологій.

Ключові слова: автоматизація медицини, система медичної статистики, електронні медичні карти та протоколи, Groovy, Grails, Hibernate, Jasper, ZK, Spring Security, DICOM.

Гоян Д.И., Краснокутская И.В. Система автоматизации учета шведско-украинского медицинского центра Angelholm. Управление и учет создают хороший имидж учреждения, предоставляющего широкий спектр медицинских услуг. Данная работа посвящается разработке программного обеспечения для клиники Angelholm с целью повышения удобства работы персонала и скорости обслуживания пациентов.

Ведение документации врачом занимает слишком много времени, он заполняет множество регистрационных форм, протоколов, карт. Предложенная нами специализированная система дает возможность легко вести полный медицинский документооборот, учет работы врачей и очереди пациентов с использованием современных информационных технологий.

Ключевые слова: автоматизация медицины, система медицинской статистики, электронные медицинские карты и протоколы, Groovy, Grails, Hibernate, Jasper, ZK, Spring Security, DICOM.

Hoyan D.I., Krasnokutska I.V. The Accounting Automation System for Swedish-Ukrainian Clinic Angelholm. Management and accounting build good impression for the organization that provides wide range of medical services. This work is devoted to software development for Clinic Angelholm with the aim to increase the convenience of staff work and the speed of patient service.

Physicians spend too much time to maintain the documentation, they have to fill out lots of registration forms, protocols, cards. Suggested specialized system gives the opportunity to maintain complete medical documentation and accounting of physicians work and patients queue without efforts and with application of up-to-date informational technologies.

Keywords: automation of health care, medical statistics system, electronic medical records and protocols, Groovy, Grails, Hibernate, Jasper, ZK, Spring Security, DICOM.

Постановка проблеми та її зв'язок із важливими практичними завданнями. Характеристикою сучасного суспільства є проникнення провідних інформаційних технологій в усі галузі людської діяльності. Медична галузь також розвивається і на стику медицини та інформаційних технологій виникають питання автоматизації роботи лікарень, аптек, діагностичних центрів та інших медичних установ.

Основною метою автоматизації є об'єднання різноманітних процесів як адміністративних, так і фінансових, як лікувальних, так і діагностичних. Автоматизацію можна порівняти із створенням єдиного інформаційного простору для організації та спрощення роботи лікарів, генерування статистики, ведення бази даних та історій хвороб пацієнтів в електронному вигляді. Це дозволяє підвищити рівень якості в загальному та ефективність кожного з наведених вище процесів. Контроль, облік та планування в єдиній системі приводять до покращення умов праці та якості послуг, що надаються. Внаслідок цього збільшується конкурентоспроможність закладу у порівнянні із іншими медичними установами.

Аналіз останніх досліджень та публікацій. Дослідження медичних інформаційних технологій є одним із ключових завдань медичної інформатики [1]. З одного боку, медичні інформаційні технології можна визначити як такі, що використовуються для обробки інформації з використанням як апаратного, так і програмного забезпечення для збереження, спільного доступу та пошуку даних з метою покращення комунікації та прийняття рішень. З іншого боку, медичні інформаційні технології можна визначити як знання, навички та інструменти, що дозволяють збирати, керувати, поширювати та використовувати інформацію для підтримки надання медичної допомоги та покращення рівня здоров'я населення [2].

В праці [3] зроблено наголос на важливості інформаційної безпеки та конфіденційності в сфері охорони здоров'я. В [4] здійснено огляд середовищ розробки мобільних додатків, використання яких надає медичному працівнику більшої мобільності. Незважаючи на численні

переваги використання інформаційних технологій в медицині, існують бар'єри на шляху до загальної комп'ютеризації цієї сфери, серед них фінансові труднощі та брак стандартів [5]. Переважна більшість досліджень в області інформаційних технологій в сфері охорони здоров'я фокусується на розробці та впровадженні програмних продуктів для обробки електронних медичних протоколів [6].

Основною метою дослідження є підвищення якості та доступності медичних послуг шляхом проектування та створення програмного продукту на основі аналізу потреб сучасної медицини. Заповнення амбулаторних карт та історій хвороб пацієнтів, формування звітів та медичної статистики не має відволікати лікаря від його основної діяльності — роботи із пацієнтами.

Виклад основного матеріалу дослідження. Медичні установи надають різноманітні за своїм характером та призначенням послуги, наприклад, різні види апаратної та лабораторної діагностики, медичні консультації, оперативні втручання. Всі вони в тій чи іншій мірі використовують інформаційні технології, мережу інтернет, електронні бази даних. Багато сучасних медичних закладів намагаються тримати руку на пульсі нових трендів і вже мають власне програмне забезпечення, написане на замовлення чи придбане через мережу. З одного боку, це може спричинити збільшення вартості послуг медичної установи, проте, з іншого боку, підвищує їх якість.

Основними перевагами використання власної спеціалізованої системи є єдина база даних та автоматизація процесів, що відбуваються в установі [7]. Єдина база даних демонструє зв'язки між фінансовими, людськими та матеріальними ресурсами. Найпростішим прикладом автоматизації процесів є планування завантаженості лікарів, діагностичного обладнання, оглядових кабінетів. За рахунок цього відбувається оптимізація використання наявних ресурсів, збільшення пропускної здатності установи, покращення якості обслуговування пацієнтів за рахунок зменшення часу очікування.

Проте, на жаль, більшість лікарень досі не мають відповідного програмного забезпечення. А спеціалізованим клінікам важко адаптувати під свої потреби існуючі програмні продукти. Процес роботи медичної установи потребує детального аналізу, оскільки кожна установа має свої вимоги до апаратного та програмного забезпечення, розподілу даних, обробки та оперування великими об'ємами цифрової інформації [7].

Клініка **Angelholm** надає широкий спектр обстежень пацієнтів від сучасної лабораторної діагностики до повного або локального обстеження із застосуванням рентгенографії, магнітно-резонансної та комп'ютерної томографії, денситометрії, ультразвукової діагностики. Під час перебування на стаціонарному лікуванні клініка забезпечує пацієнта якісним лікуванням, індивідуальними методиками реабілітації, висококваліфікованим доглядом. Завдяки 25-річному досвіду, наполегливій роботі, новим європейським технологіям, використанню сучасних медикаментів і матеріалів у клініці успішно проводиться гарантоване хірургічне лікування в поєднанні з індивідуальним підходом до кожного пацієнта [8]. З метою розвитку та підвищення якості шведсько-український медичний центр **Angelholm** потребує хорошого програмного забезпечення та автоматизації його роботи.

Результатом даного дослідження є програмний продукт, впроваджений у клініці. Він бере на себе облік усіх послуг, що надаються. Система медичного керування дозволяє працівникам організації на реєстратурі створювати електронні амбулаторні картки, вести облік медичних книг, записувати пацієнтів на прийом. Пацієнти мають можливість проводити оплату із використанням онлайн-банкінгу. За допомогою такого програмного забезпечення можна відслідковувати передоплату та заборгованість клієнтів, готівкові та безготівкові розрахунки за надані послуги. Медичний персонал може бачити список пацієнтів, записаних до них на прийом на поточний день. Доступ до будь-яких даних може відбуватися із робочого місця лікаря, що спрощує медичний облік в разі. Програма формування заключення після огляду легко налаштовується під будь-якого працівника, кожен лікар може створювати свої шаблони для заповнення карти прийому чи огляду пацієнта.

Для забезпечення виконання поставленої задачі був проведений аналіз засобів програмування і для реалізації вибрані наступні технології, які останнім часом стали досить популярними. При написанні програмного проекту використано скриптову мову програмування **Groovy**. Вона розроблена для **платформи Java** і є більш високорівневою мовою програмування порівняно з **Java**, розробка на ній зазвичай відбувається швидше за рахунок динамічної природи

мови та елементів функціонального програмування [9]. В якості основи системи взято фреймворк **Grails** — високо-продуктивний програмний каркас для створення веб-додатків для платформи **Java**, створений під сильним впливом широко відомого **Ruby on Rails** [10] і заснований на архітектурному шаблоні проектування програмного забезпечення **MVC** (Модель – Представлення – Контролер) [11]. Для розміщення даних вибрана система керування реляційними базами даних **MySQL**, що широко використовується для створення динамічних веб-сторінок і підтримується багатьма мовами програмування [12]. Комунікація між проектом та базою даних виконана за допомогою інструменту **Hibernate**, що призначений для розв'язування задач об'єктно-реляційного відображення [13]. Для генерації **PDF**-заключень використовується **Java**-бібліотека **JasperReports**, що генерує звіти на основі **XML**-шаблонів [14]. Для створення користувацького інтерфейсу використовується фреймворк **ZK**, розмітка знаходиться у **zul**-файлах, що інтерпретуються в **html**. За захист відповідає фреймворк **Spring Security**, що надає механізми побудови систем аутентифікації та авторизації в **Java**-додатках [15]. Всі вище перелічені інструменти підключаються до проекту як плагіни.

Працювати в програмі можуть тільки авторизовані користувачі. Усі паролі хешуються. **Spring Security** використовує такі класи як **User**, **Role** і **UserRole**. Кожен медичний працівник має відповідну роль: адміністратор, лікар, працівник реєстратури чи колцентру. При роботі із певною функціональністю програма визначає рівень доступу користувача і перелік функцій, якими він може користуватися.

В проекті реалізована можливість працювати із графіком роботи лікаря за допомогою календаря та налаштовувати зручний для нього візуальний вигляд. Працівники клініки можуть записувати пацієнтів на певний вид обстеження на вибраний час, проводячи пошук пацієнтів в базі. Для зручності пошуку медичні працівники можуть використовувати відповідні фільтри, наприклад пошук за пацієнтом, типом обстеження, датою тощо. Ключовою особливістю даного розділу є фіксація усіх змін і можливість блокувати дні чи години, коли лікар зайнятий і не проводить обстеження.

Іншою, не менш важливою особливістю продукту, є можливість створювати електронні платежі через інтеграцію із системою Приват24. Для генерування платежу потрібна інформація отримується з бази даних, а платіж, в свою чергу зберігається у профілі відповідного пацієнта. Пацієнт може провести оплату через термінал. Також дана програма формує звіти за платежами за вказаний період чи за виконаними обстеженнями або маніпуляціями.

DICOM — це галузевий стандарт створення, збереження, передачі та візуалізації медичних зображень та протоколів обстежень пацієнтів, що підтримується основними виробниками медичного обладнання та медичного програмного забезпечення [16]. Деякі види медичного обладнання, наприклад апарати МРТ, працюють із **DICOM** сервером, вони мають свою консоль із відповідним інтерфейсом. Спочатку необхідно ввести дані про пацієнта й обстеження, потім провести маніпуляцію і після цього вихідні дані потраплять на **DICOM** сервер. Файли обстежень зазвичай мають формат **dcm**, вони являють собою об'єктно-орієнтовані файли з теговою організацією.

У даному програмному продукті при обстеженні пацієнта його дані отримуються із бази, формується **worklist** і відправляється на **DICOM** сервер. При пошуку пацієнтів в базі лікар може користуватися фільтрами пошуку, що значно прискорить його роботу. Для генерування протоколу обстеження використовується спеціальний розділ, де лікар-діагност на основі знімків і огляду пацієнта формує заключення за допомогою шаблонів і готових фрагментів тексту. Поточне заключення закріплюється за лікарем, який провів обстеження та сформував протокол, інші лікарі можуть тільки переглядати і друкувати його. Якщо лікар вважає за необхідне направити пацієнта на подальші обстеження після огляду, він може це зробити в розділі опису заключень, при цьому відповідні скерування одразу фіксуються в базі.

Інший розділ програми надає можливість генерувати етикетки-наклейки для конвертів, в яких пацієнту повертають результати обстеження. При наявності спеціального принтера, етикетку можна роздрукувати на поверхню диску, що дозволяє такий друк, на якому записані результати обстеження.

Завдання керування чергою пацієнтів вимагає особливої уваги. Облік черги пацієнтів реалізований в окремому розділі проекту і може транслюватися у приймальні на моніторах або телевізорах, підключених до мережі. Лікар може вказувати можливість прийому на свій сеанс, не виходячи з кабінету. Відображається інформація про зайнятість кабінетів та лікарів і список

пацієнтів, що очікують своєї черги. Для повідомлень щодо зміни стану черги використовується сповіщення із текстовим і звуковим сигналами для привернення уваги пацієнтів.

Висновки. В процесі проведення дослідження здійснено огляд сучасних медичних інформаційних технологій та їхніх можливостей при обробці відповідної біомедичної інформації. Програмне забезпечення, спроектоване та розроблене в процесі дослідження, пройшло етап тестування та впроваджене в шведсько-українському медичному центрі **Angelholm** і наразі успішно виконує свої функції.

Програма медичного контролю формує єдину базу всіх пацієнтів, що звертаються на огляд або діагностику. Медичне керування включає в себе попередній запис до будь-якого лікаря або в діагностичний кабінет. Автоматизоване ведення справ містить журнал медичного обліку. Система медичної реєстрації містить всі обстеження даного пацієнта та результати консультативних прийомів.

Відповідно до потреб клініки в проекті реалізовані розділи роботи з календарем, сумісної роботи з **DICOM**-сервером, формування заключень та протоколів обстежень, друком етикеток та дисків, чергою пацієнтів.

Керування прийняттям рішень в медицині, використання програм медичного обліку, автоматизація роботи лікарів в лікувальних та діагностичних медичних закладах — все це не лише дуже зручно, а і є показником рівня медичного закладу, що формує відношення пацієнтів та думку інших колаборуючих організацій.

Перспективи подальших досліджень. Для подальшого розвитку проекту та підвищення зручності роботи працівників клініки можна врахувати актуальні потреби сфери охорони здоров'я та доповнити дане програмного забезпечення:

- обліком складу медикаментів;
- електронною медичною бібліотекою;
- додатком цільового планування, що дозволить закладати бюджет на кожне відділення медичного закладу;
- взаємодією із страховими компаніями;
- експертною системою, ефективною при вирішенні задач діагностики, інтерпретації даних, прогнозуванні перебігу захворювань і ускладнень, моніторингу перебігу захворювань і планування лікувально-діагностичного процесу;
- мобільним додатком.

1. Hersh William Medical informatics: improving health care through information / Hersh William // The Journal of the American Medical Association. — 2002. — 288(16). — P. 1955–1958 <http://dx.doi.org/10.1001/jama.288.16.1955>
2. Thompson T. G. The decade of health information technology: delivering consumer-centric and information-rich health care: framework for strategic action / T. G. Thompson, D. J. Brailer. — Washington: D.C. — 2004. — 13 p.
3. Appari Ajit Information security and privacy in healthcare: current state of research / Ajit Appari, Eric Johnson // International Journal of Internet and Enterprise Management. — 2010. — 6, N 4. — P. 279–314. <http://dx.doi.org/10.1504/IJIEEM.2010.035624>
4. Ehrler Frederic Technological choices for mobile clinical applications / Frederic Ehrler, David Issom, Christian Lovis // Studies in Health Technology and Informatics. — 2011. — 169. — P. 83–87. <http://dx.doi.org/10.3233/978-1-60750-806-9-83>
5. Bates David Improving Safety with Information Technology / David Bates, Atul Gawande // The New England Journal of Medicine. — 2003, 358. — P. 2526–2534. <http://dx.doi.org/10.1056/NEJMsa020847>
6. Holdena Richard The Technology Acceptance Model: Its past and its future in health care / Richard Holdena, Ben-Tzion Karsha // Journal of Biomedical Informatics. — 2010. — 43, N 1. — P.159–172. <http://doi.org/10.1016/j.jbi.2009.07.002>
7. Скорін Ю.І. Впровадження інформаційного супроводу пацієнтів / Ю.І. Скорін, А.І. Пугачов, О.В.Щербаков // Збірник наукових статей "Управління розвитком". — 2012. — 4 (102), 1. — с. 59–61.
8. Шведсько-українського медичного центру Angelholm [Електронний ресурс] — Режим доступу до ресурсу: <http://angelholm.com.ua/>
9. Koenig Dierk Groovy in Action / Dierk Koenig, Paul King. — Manning Publications. — 2015. — 912 p.
10. Beckwith Burt Programming Grails. Best Practices for Experienced Grails Developers / Burt Beckwith. — O'Reilly Media. — 2013. — 364 p.
11. Знайомство із MVC через приклади на PHP [Електронний ресурс] — Режим доступу до ресурсу: http://isblog.ua/2013-08_mvc-pattern-and-php
12. Pipes Jay Pro MySQL / Jay Pipes, Michael Kruckenberg. — Apress. — 2005. — 768 p.
13. Mak Gary Hibernate Recipes. A Problem-Solution Approach / Gary Mak, Srinivas Guruzu, Joseph Ottinger. — Apress. — 2015. — 284 p.
14. Siddiqui Bilal JasperReports 3.6. Development Cookbook / Bilal Siddiqui. — Packt Publishing. — 2010. — 396 p.
15. Gutierrez Felipe Pro Spring Boot / Felipe Gutierrez. — Apress. — 2016. — 365 p.
16. Noumeir Rita Benefits of the DICOM Structured Report / Rita Noumeir // Journal of Digital Imaging. — 2006. — 19 (4). — P. 295–306. <http://dx.doi.org/10.1007/s10278-006-0631-7>

УДК УДК 004.413 (045)

Жигаревич О.К., Котлярець В.В., Лець А.Р.

Луцький національний технічний університет

МОДЕЛЬ ЕКОСИСТЕМИ НАВЧАЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Жигаревич О.К., Котлярець В.В., Лець А.Р. Модель екосистеми навчального програмного забезпечення.

Розробка програмного забезпечення включає в себе соціально – технічні системи. Сучасне програмне забезпечення характеризується масивністю, складністю, швидкою інтеграцією у соціальні відносини, застосуванням нових принципів при формулюванні задачі, проектуванні програмного забезпечення, розробці, супроводі, обміні та використанні. Саме взаємодія різних сфер впливу на розробку програмного забезпечення спонукала до застосування нових принципів при створенні моделі екосистеми навчального програмного забезпечення.

Ключові слова: модель, екосистема, програмне забезпечення, дослідження, літературні джерела.

Жигаревич О.К., Котлярець В.В., Лець А.Р. Модель экосистемы учебного программного обеспечения.

Разработка программного обеспечения включает в себя социально – технические системы. Современное программное обеспечение характеризуется массивностью, сложностью, быстрой интеграцией в социальные отношения, применением новых принципов при формулировке задачи, проектировании программного обеспечения, разработке, сопровождении, обмене и использовании. Именно взаимодействие разных сфер влияния на разработку программного обеспечения побудило к применению новых принципов при создании модели экосистемы учебного программного обеспечения.

Ключевые слова: модель, экосистема, программное обеспечение, исследования, литературные источники.

Zhyharevych O.K. Kotlyarets V.V. Lets A.R. The model of educational software ecosystem. Software development includes socio - technical systems. Modern software is characterized by its massiveness, complexity, rapid integration into social relations, the use of new principles in the formulation of the problem, software design, development, maintenance, sharing and use. It is the interaction of different spheres of influence in software development led to the use of new principles in creating a model of educational software ecosystem.

Keywords: model, ecosystem, software, research, literature.

Вступ

Програмне забезпечення швидко розвивається і поширюється у різних сферах життя сучасного українця. Кожна друга людина не може себе уявити без технічного пристрою, який допоможе знайти потрібну інформацію на пошукових сайтах, переглянути новини, набути знань у тій чи іншій сфері життя. Технічний прогрес включає в себе і прогрес програмного забезпечення, який народжується, розвивається, перебігає, живе в певному середовищі. Процес поєднання програмного забезпечення і суспільства викликає потребу у використанні додаткових понять і концепцій для вивчення і опису програмного забезпечення. Одним із методів вирішення даної проблеми є використання концепції екосистем програмного забезпечення. Саме тому екосистеми програмного забезпечення можуть стати додатковим інструментом для дослідження навчального програмного забезпечення.

Аналіз останніх досліджень. За останні кілька років поняття «екосистема програмного забезпечення» і супутні терміни активно використовуються розробниками і дослідниками програмного забезпечення. Огляд веб-сайтів провідних розробників програмного забезпечення показує, що більшість з них застосовують поняття «екосистема програмного забезпечення» (наприклад [1-4]), позначаючи ним системи, що включають підприємство розробника, його програмне забезпечення і партнерів. Розробники застосовують цей термін «як є», без теоретичних бази або посилань на відповідні праці. Деякий виняток становить корпорація «Майкрософт», яка дає визначення екосистеми програмного забезпечення як сукупності взаємодій і взаємовпливів організацій (державних, навчальних і комерційних) і індивідумів, що працюють із програмним забезпеченням [4].

Наукові дослідження, які використовують поняття екосистем, на сьогоднішній день представлені кількома працями [6-8]. В роботі [6] «Екосистеми програмного забезпечення» автори описують типові елементи екосистем та їх контекст, роблячи спробу на якісному рівні спрогнозувати характеристики розвитку глобальної екосистеми програмного забезпечення з точки

зору подальшого підвищення ефективності технологій розробки, появи і розвитку нових областей застосування.

Дослідження екосистеми навчального програмного забезпечення, на даний час є дуже актуальним. Використання та дослідження навчального програмного забезпечення у вищих навчальних закладах України дає багато переваг для оволодіння тією чи іншою професією, а пізнання, засвоєння інформації в певному середовищі і породжує модель екосистеми навчального програмного забезпечення.

Структура web-додатку

На рисунку 1 представлено порядок дій викладача для створення власної системи навчання і оцінювання студента, а також порядок дій студента для отримання кінцевого результату. Наповнення навчальною інформацією бази знань, представлення інформації для навчання, програми для тестування перевірки знань на різних освітніх рівнях.[1]

Для створення системи по навчанню і оцінюванню студента викладачу потрібно зареєструватись на сайті та отримати підтвердження акаунту від адміністратора або редактора сайту. Для цього на сайті існує сторінка з формою де користувач вказує унікальне ім'я, імейл, пароль, ім'я, прізвище, по-батькові, а в кінці зазначає тип ролі. В нашому випадку - викладач. Всі поля - є обов'язковими для заповнення.

The screenshot shows a web browser window with the 'Courses' website. The page title is 'User account'. The form is titled 'User account' and has a 'Create new account' button. The form fields are:

- Username ***: A text input field containing 'teacher'.
- E-mail address ***: A text input field containing 'teacher@teacher.ua'.
- Password ***: A text input field with a password strength indicator showing 'Weak'.
- Confirm password ***: A text input field with a 'Passwords match: yes' status.
- First name ***: A text input field containing 'First teacher name'.
- Middle name ***: A text input field containing 'Middle teacher name'.
- Last name ***: A text input field containing 'Last teacher name'.
- Role ***: A dropdown menu with 'student' and 'teacher' options. 'teacher' is selected.

At the bottom of the form is a 'Create new account' button.

Courses Головна

Обліковий запис користувача / Реєстрація

Обліковий запис користувача

Регистрація Вхід Залит нового паролю

Ім'я *

Прізвище *

По батькові *

Ім'я користувача *

Електронна адреса *

Дійсна адреса електронної пошти. Усі повідомлення з сайту будуть надсилатися на цю адресу. Її не буде оприлюднено та буде використано лише за Вашим бажанням для отримання нового паролю або новин чи сповіщень електронною поштою.

Пароль *

Підтвердити пароль *

Регистрація

Рисунок 1 – Обліковий запис користувача

Для створення системи по навчанню і оцінюванню студента, викладачу потрібно зареєструватися на сайті та отримати підтвердження акаунту від адміністратора або редактора сайту. Для цього на сайті існує сторінка з формою де користувач вказує унікальне ім'я, електронну адресу, пароль, ім'я, прізвище, по-батькові. Всі поля - є обов'язковими для заповнення. Крім ролей "Студента" і "Викладача" також існують ролі такі, як "Адміністратор", "Редактор". Одною із обов'язків яких є додавання студентів на сайт. Для цього вони можуть використати сторінку "Додавання студентів" на якій розміщена форма з полям вибору кількості додавання користувачів і таблицею з полями студентів.

Панель керування Вийти Групи Структура Вигляд Люди Модулі Quiz Конфігурація Зали Додати admin Вийти

Courses Головна Курси Студенти Мій обліковий запис Вийти

Додавання студентів

Кількість рядків

10

#	Електронна пошта	Серія, номер	Ім'я	Прізвище	По батькові	Номер телефону
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

Ok

Рисунок 2 – Додавання студентів

Після створення студента адміністративні ролі можуть змінювати дані студента на сторінці редагування користувача.

The screenshot shows a web interface for editing a student's profile. At the top, there is a navigation bar with 'Courses', 'Головна', and 'Курси'. On the right, there are links for 'Мій обліковий запис' and 'Вийти'. Below the navigation bar, there is a breadcrumb trail: 'Student name 1 / Редагування'. The main heading is 'Student name 1'. There are two tabs: 'Перегляд' (View) and 'Редагування' (Edit), with 'Редагування' being the active tab. The form contains several input fields: 'Ім'я *' (Name) with 'Ім'я 1', 'Прізвище *' (Surname) with 'Прізвище 1', 'По батькові *' (Patronymic) with 'По батькові 1', 'Серія, номер' (ID Number) with 'ФП0133013', and 'Номер телефону' (Phone Number) with a placeholder '+'. There is a blue button '+ Додати елемент' (Add element) next to the phone number field. Below the phone number field, there is a section for 'Діючий пароль' (Active password) with a text input field and a link 'Введіть діючий пароль, щоб змінити Електронна адреса або Пароль. Запит нового паролю.' (Enter active password to change email or password. Request new password.). Below this is a section for 'Електронна адреса *' (Email) with a text input field containing 'student@example.com' and a link 'Діюча адреса електронної пошти. Усі повідомлення з сайту будуть надіслані на цю адресу. Її не буде оприлюднено та буде використано лише за Вашим бажанням для отримання нового паролю або новин чи сповіщень електронною поштою.' (Active email address. All messages from the site will be sent to this address. It will not be published and will be used only if you wish to receive a new password or news or notifications by email.). Below the email field, there are two more text input fields for 'Пароль' (Password) and 'Підтвердити пароль' (Confirm password). At the bottom, there is a green button 'Зберегти' (Save).

Рисунок 3 – Авторизація користувача-студента

Після підтвердження аккаунта редактором, викладач, зайшовши на сайт має три основних посилання:

- головна сторінка,
- сторінка груп викладача,
- сторінка всіх студентів сайту.

Першим кроком для створення процедури з навчання студента є додавання курсу. Для цього викладачу потрібно перейти на сторінку з списком курсів, де доступна опція для створення нового курсу.

The screenshot shows a web interface for the 'Courses' section. At the top, there is a navigation bar with 'Courses', 'Головна', 'Курси', and 'Студенти'. On the right, there are links for 'Мій обліковий запис' and 'Вийти'. Below the navigation bar, the main heading is 'Курси'. There is a blue button '+ Створити курс' (Create course) and a link 'Курс 1'.

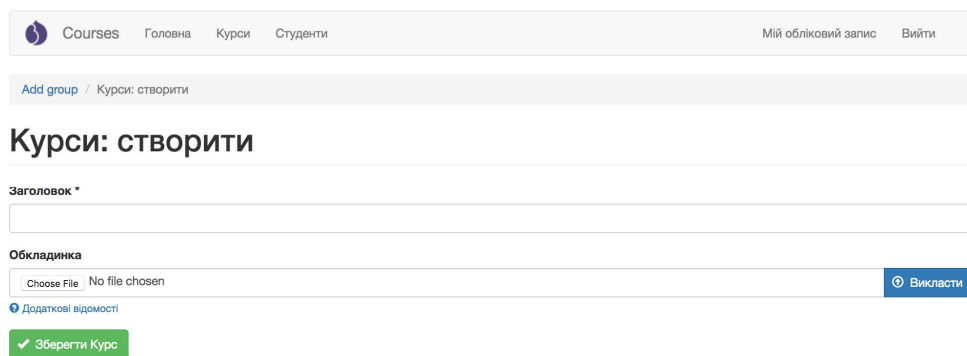


Рисунок 4 – Створення нового курсу

Викладач дає назву своєму курсу, це поле є обов'язковим і може завантажити обкладинку. Далі клікає на кнопку “зберегти курс”. Після збереження викладач потрапляє на сторінку перегляду курсу. На сторінці перегляду знаходяться посилання на налаштування курсу, графіки проходження тестів і список студентів. Список студентів можна експортувати в CSV файл.[2]

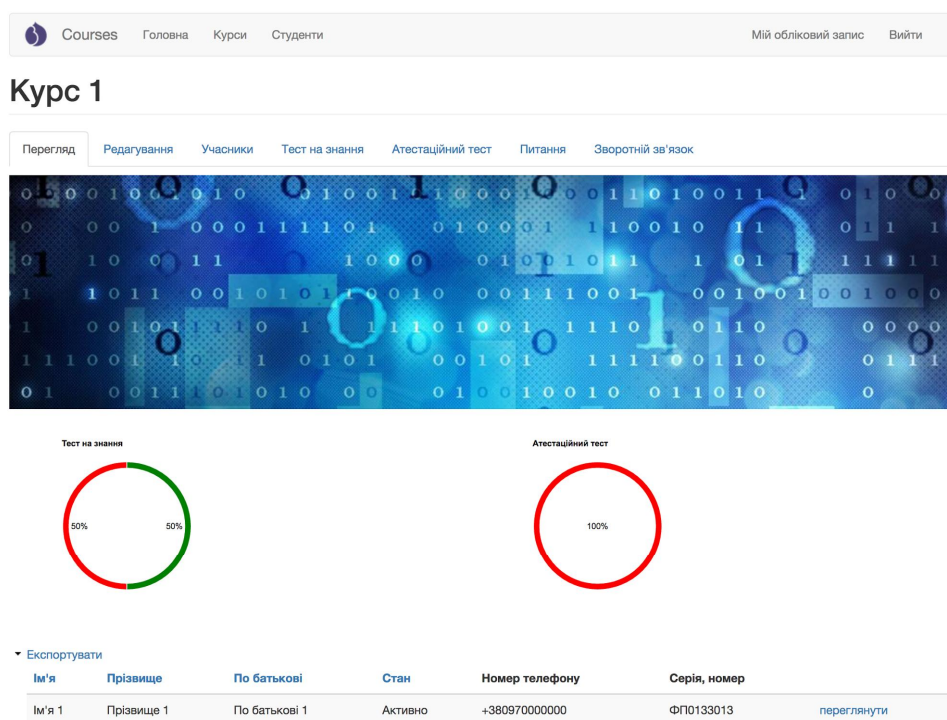


Рисунок 5 – Ноповнення навчального курсу

Крім того готовий курс можна редагувати, змінити заголовок та обкладинку курсу.

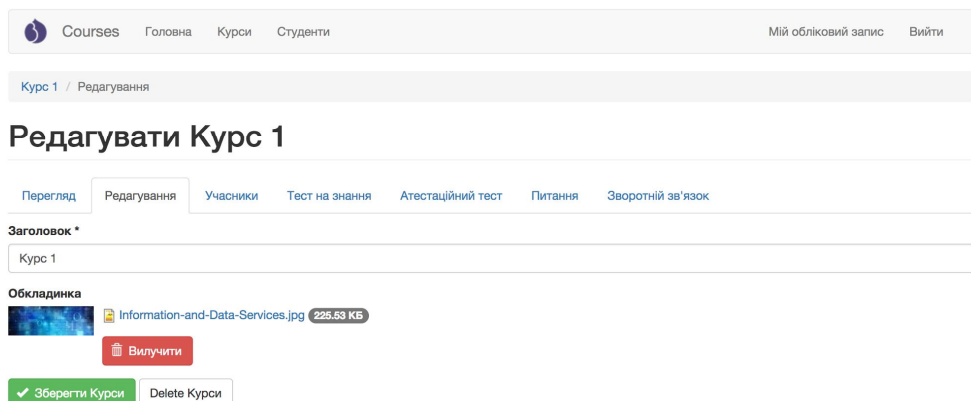


Table 1

Ім'я	Прізвище	По батькові	Стан	Номер телефон	Серія, номер
Ім'я 1	Прізвище 1	По батькові 1	Активно	+380970000000	ФП0133013

Рисунок 6 – Редагування курсу

Кожен курс з боку викладача складається з:

- вступного тесту для студента, щоб визначити його початковий рівень знань;
- атестаційного тесту для контролю знань після пройденого курсу;
- зворотнього зв'язку з рекомендаціями від викладача;
- питань поточної групи;
- списку студентів, які проходять курс;

Після того як курс був створений, викладачу потрібно налаштувати процедуру проходження тестів. Для цього йому потрібно перейти на один з доступних сторінок тесту на якій доступні наступні опції:

- кількість випадкових питань;
- статус тесту;
- доступний таймер;
- час проходження тесту;

На сторінці налаштування тесту (як на знання, так і атестаційного) можна:

- редагувати список питань;
- експортувати список результатів в CSV файл;
- змінити заголовок тесту;
- опублікувати/не опублікувати тест;
- виключити/налаштувати таймер;

Courses
Головна
Курси
Студенти
Мій обліковий запис
Вийти

Курс 1

Перегляд
Редагування
Учасники
Тест на знання
Атестаційний тест
Питання
Зворотній зв'язок

Питання

Операції
- Оберіть операцію - Виконати

Не знайдено

Результати

Експортувати
Операції
- Оберіть операцію - Виконати

Серія, номер	Ім'я	Прізвище	По батькові	Результат	
ФП0133013	Ім'я 1	Прізвище 1	По батькові 1	10	

Діаграми

Налаштування

Заголовок *
Skill test
Опубліковано
Включити таймер
Хвилини
10
Зберегти

Рисунок 7 – Авторизація користувача

Якщо потрібних питань до курсу не було в списку, тоді для додавання нових, користувачу доступна окрема сторінка для створення питання і список всіх доступних питань.

Кожне запитання складається з заголовка (може бути просто вказаний послідовний номер), самого формулювання запитання та відповіді.[3]

Після створення – викладач має змогу проводити наступні операції над питаннями:

- опублікувати питання з курсу;
- видалити питання з курсу;
- скасувати публікацію курсу на сайті;

Перед тим як приєднати питання до певного курсу, викладач повинен поставити прапорець навпроти вибраних запитань.[4]

Courses
Головна
Курси
Студенти
Мій обліковий запис
Вийти

Курс 1

Перегляд
Редагування
Учасники
Тест на знання
Атестаційний тест
Питання
Зворотній зв'язок

Створення запитання

Операції
- Оберіть операцію - Виконати

Заголовок	Стан	
124124	Опубліковано	
Тест	Опубліковано	

Рисунок 8 – Перевірка знань користувача

На сторінці учасників курсу доступний всіх членів курсу, в тому числі викладачі. Над учасниками можна виконувати наступні операції:

- змінювати статус (активний/неактивний);
- фільтрувати та сортувати список за статусами та ролями;
- додавати/видаляти ролі;

Рисунок 9 – Внесення поправок

Сторінка зворотнього зв'язку використовується викладачами для надання рекомендації щодо проходження курсу. На даній сторінці знаходиться форма, де вказується відсоток успішності проходження тесту і рекомендовані джерела інформації в залежності від набраного балу.

CoursesГоловнаКурсиСтудентиМій обліковий записВийти

Курс 1

ПереглядРедагуванняУчасникиТест на знанняАтестаційний тестПитанняЗворотній зв'язок

Від *

%

До *

%

Body

Від *

%

До *

%

Body

Від *

%

До *

%

Body

Від *

%

До *

%

Body

Від *

%

До *

%

Body

Зберегти

Попередній перегляд

Рисунок 9 – Загальний вигляд вікна перевірки знань студентів

Висновок

Модель екосистема навчального програмного забезпечення представляє собою взаємозв'язок розробників програмного забезпечення, користувачів програмного забезпечення, та навчальну установу ВУЗ. Потрібно розуміти, що ключову роль у цій взаємодії відіграє людина, бо вона розробляє, користується, та навчається. Можна говорити про екосистему навчального програмного забезпечення, як одну із галузей науки, яку потрібно, ще довго досліджувати. Практичним результатом досліджень екосистеми навчального програмного забезпечення є побудова моделі екосистеми навчального програмного забезпечення та реалізація її за допомогою web – додатку.

1. Approach.[Text]/ S. Jansen, J. Bosch.// Proceedings of the Workshop on Software Ecosystems – 2011. – P. 65 – 76.
2. Duinkerken W. Transaction Cost Economics in Software Ecosystems. [Text] // Some empirical evidence. – April 20, – 2009. – P. 22 - 37.
3. Dyba T. Empirical studies of agile software development . [Text] / T.Dyba T.Dingsoyr // A systematic review. Information and Software Technology. – 2008, vol. 50, pp. 833—859.
4. Messershmitt D. Ultra-Large-Scale Systems.[Text] / D. Messershmitt, C. Szyperski.// Understanding an Indispensable Technology and Industry. – London: MIT press, – 2003. – 233 p.
5. Сидоров М.О. Программное обеспечение – экологический подход к исследованиям. [Текст] // Інженерія програмного забезпечення. – Київ, – 2010. – № 1. – С. 5 – 13.
6. Сидоров Н.А. Экология программного обеспечения. [Текст]//Інженерія програмного забезпечення. – Київ, – 2010. – № 1. – С. 53 – 61.
7. Хоменко В.А. «Екосистеми програмного забезпечення» [Текст] // Вісник Національного технічного університету «ХПІ». – Харків, – 2011. – № 23. – С. 114 – 118.
8. Луцький М. Підтримка придатності та супроводження експлуатації програмного забезпечення авіаційної техніки./ М.Луцький М. Сидоров, Ю. Рябокінь. [Текст] // – Проблеми програмування. – №23. – Київ. – 2010. – с. 229 – 239.

УДК 621.3

Н.В. Здолбіцька, А.П. Здолбіцький

Луцький національний технічний університет

ПРИСТРІЙ ПІДВИЩЕННЯ ЕНЕРГОЕФЕКТИВНОСТІ СИСТЕМИ КЕРУВАННЯ ЕЛЕКТРОМАГНІТНИМИ КЛАПАНАМИ

Н.В. Здолбіцька, А.П. Здолбіцький. Пристрій підвищення енергоефективності системи керування електромагнітними клапанами. Пристрій призначений для підвищення енергоефективності існуючих промислових систем керування електромагнітними клапанами автомобільного газобалонного обладнання. В основі роботи пристрою лежить спосіб імпульсного відкриття електромагнітних соленоїдальних клапанів, із наступним утриманням їх в робочому стані зниженням струмом та напругою.

Ключові слова: електромагнітний клапан, стабілізатор струму, енергоефективність.

Рис. 3. Літ. 12.

Н.В. Здолбицкая, А.П. Здолбицкий. Устройство повышения энергоэффективности системы управления электромагнитными клапанами. Устройство предназначено для повышения энергоэффективности существующих промышленных систем управления электромагнитными клапанами автомобильного газобаллонного оборудования. В основе работы устройства лежит способ импульсного открытия электромагнитных соленоидальных клапанов, с последующим удержанием их в рабочем состоянии пониженным током и напряжением.

Ключевые слова: электромагнитный клапан, стабилизатор тока, энергоэффективность.

N.V. Zdolbitska, A.P. Zdolbitsky. The device is energy efficiency management system solenoid valve. The device is designed to improve the energy efficiency of existing industrial control systems electromagnetic valves automotive LPG equipment. At the core of the device is the way to opening pulse electromagnetic solenoid valve, followed by keeping them in working order low current and voltage.

Keywords: solenoid valve, a current, energy efficiency.

Постановка задачі. Проблема побудови енергоефективних систем керування електромеханічними системами є досить актуальною, особливо для випадку їх застосування у складі автономних та мобільних систем, де є обмеження на тривалість роботи та споживану потужність.

Відомо, що струм відпускання якоря електромеханічного клапана менший, ніж струм спрацьовування. Робочий струм соленоїдального електромагнітного клапана в технічній документації вказують з допустимими межами, в яких гарантована надійна робота клапана і він повинен бути завжди більшим, ніж струм спрацьовування. Нижнє значення струму дорівнює добутку струму спрацьовування на деякий мінімальний коефіцієнт запасу. Запас забезпечує необхідний час спрацьовування та характеризує працездатність клапана при впливі дестабілізуючих факторів [1,7].

Для утримання електромагнітного клапана в робочому положенні (при відсутності ударів і вібрацій) необхідний струм утримання зазвичай менший робочого в декілька разів. Звідки, для підвищення енергоефективності системи штучно обмежують струм через обмотку електромагнітного клапана після його спрацьовування.

Отже, актуальною є задача створення пристрою, що забезпечує живлення соленоїдів клапанів газобалонного обладнання (ГБО) із значно меншими енергозатратами, без втрати функціонування та захистом від імпульсів зворотної ЕРС під час перехідних станів відкриття та закриття клапанів.

Поставлена задача вирішується тим, що принцип дії поширених на ринку автомобільних електромагнітних газових клапанів та регуляторів для зрідженого нафтового газу LPG (пропан-бутанові суміші) та природного газу LNG (метану) базується на властивості втягування металевго осердя соленоїдом [1,8], а це дозволяє утримувати їх в робочому (відкритому) стані за допомогою значно меншої напруги та сили струму, ніж необхідно для їх початкового відкриття [2]. Тому використання схеми імпульсного відкриття клапанів ГБО, із наступним утриманням їх в робочому стані зниженим струмом (напругою), дозволяє підвищити енергоефективність системи, знизити навантаження на електропроводку та генератор авто, і, як наслідок, отримати економію палива. До того ж, використання інтегральних стабілізаторів та шунтуючих діодів для соленоїдів клапанів, дозволяє подолати імпульси зворотної ЕРС під час перехідних процесів у системі.

Аналіз останніх досліджень і публікацій. Відомий спосіб імпульсного вмикання реле струмом розряду конденсатора [2, 6], який підключається паралельно до обмотки вихідного реле і

заряджається під час спрацювання командного реле, а розряджається на обмотку вихідного реле після відключення командного реле.

До недоліків даного способу можна віднести короткотривале вмикання реле, час вмикання якого залежить від ємності конденсатора та обмежувального опору.

Відомий також пристрій для керування електромагнітом постійного струму [3], який містить ключі керування, дросель, конденсатор, блоки керування, джерело напруги, діоди.

Недоліком даного пристрою є потреба у двох джерелах живлення, що призводить до підвищеного енергоспоживання та складності пристрою.

Найбільш близькими за технічною суттю до пристрою, що пропонується, є спосіб керування електромагнітними механізмами, зокрема електромагнітними газовими клапанами, та пристрій для його реалізації [4] з одним основним і одним додатковим джерелами напруги постійного струму.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Пристрій містить основне джерело напруги постійного струму, додаткове джерело напруги постійного струму, що складається з зарядного пристрою і накопичувального конденсатора, і, відповідно до кількості електромагнітних механізмів, – блоки управління, кожен з яких містить два керувані ключі зі своїми вузлами керування і таймер. Командні сигнали надходять на вузли управління кожного електромагніту по своєму провіднику, причому на вузол керування другого ключа – через таймер. Між джерелами напруги і обмоткою керованого електромагнітного механізму включені розділюючі діоди. Шунтуючі діоди включені паралельно керованим ключам та обмотці електромагніту. Таймер задає час вмикання електромагнітного механізму.

У даному способі керування електромагнітним механізмом при надходженні керуючого сигналу його включення спочатку подають напругу від двох джерел напруги постійного струму – від основного і додаткового, а після деякої затримки в часі, достатньої для включення електромагніту в роботу, припиняють подачу напруги від додаткового джерела напруги постійного струму і утримують електромагніт в робочому стані шляхом подачі напруги тільки від основного джерела напруги постійного струму, для вимкнення електромагнітного механізму припиняють подачу напруги від основного джерела напруги постійного струму.

Недоліком даного пристрою є потреба у зарядному пристрої для додаткового джерела напруги, а також необхідність таймерів для кожного вузла керування, що значно ускладнює пристрій.

В основу проектного пристрою поставлена задача розробити енергоефективний спосіб підключення системи керування клапанами автомобільного газобалонного обладнання.

Поставлена задача вирішується шляхом використання схеми імпульсного відкриття клапанів газобалонного обладнання за допомогою розряду накопичувального конденсатора великої ємності на котушки газових клапанів, із наступним утриманням їх в робочому стані зниженим стабілізованим струмом, оскільки процес утримання газових клапанів у робочому положенні потребує значно меншої напруги та сили струму, ніж необхідно для їх початкового відкриття, що дозволяє підвищити енергоефективність системи, знизити навантаження на електропроводку та генератор авто, і, як наслідок, отримати економію палива. Накопичувальний конденсатор заряджають через розділюючий діод від джерела живлення в режимі «бензин». Для підвищення енергоефективності в режимі роботи «газ» комутуюче реле вмикають так, щоб воно не споживало струм утримання. Живлення котушок газових клапанів зниженим струмом дозволяє уникнути їх надмірного перегріву, а застосування інтегральних стабілізаторів струму дозволяє уникнути наслідків їх короткого замикання. Захист бортової мережі автомобіля від імпульсів зворотної електрорушійної сили під час перехідних станів відкриття та закриття газових клапанів досягається шляхом введення у схему захисних шунтуючих діодів.

Суть проектової системи пояснюється кресленням, на якому представлена схема електрична принципова пристрою (рис. 1).

Підвищення енергоефективності системи керування електромагнітними клапанами автомобільного газобалонного обладнання здійснюють за допомогою пристрою, який містить вхід живлення для режиму «газ» 1, вхід живлення для режиму «бензин» 2, розділюючий діод 3, накопичувальний конденсатор 4, комутуюче реле 5, шунтуючі діоди 6 та 7, котушку газового клапана 8, стабілізатор струму 9, «масу» автомобіля 10, розділюючий діод 11, котушку газового клапана 12, стабілізатор струму 13, шунтуючий діод 14, розділюючий діод 15, котушку газового клапана 16, стабілізатор струму 17, шунтуючий діод 18.

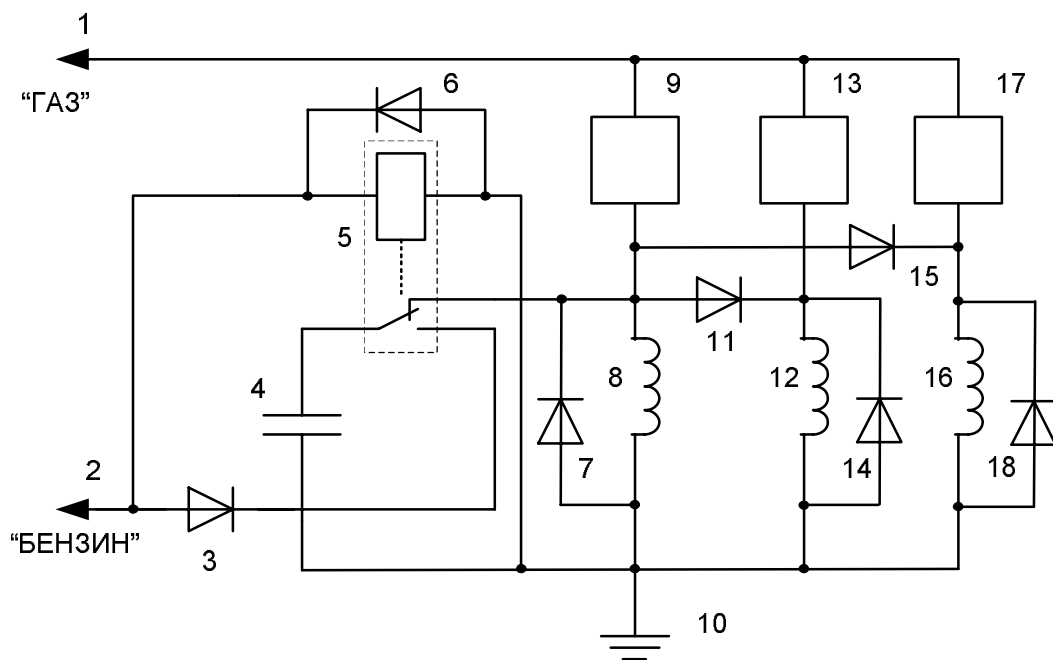


Рисунок 1 – Схема електрична принципова пристрою

На кресленні зображено варіант схеми імпульсного запуску клапанів газобалонного обладнання легкового автомобіля, в якому керуючі сигнали вибору режиму роботи «газ/бензин» об'єднані із основним джерелом живлення +12 В, яке послідовно комутується до входу живлення для режиму «бензин» 2, або до входу живлення для режиму «газ» 1, в залежності від виду палива на якому працює двигун автомобіля. Запропонований пристрій може керувати трьома котушками газових клапанів. Кількість котушок може варіюватися в залежності від модифікації та покоління газобалонного обладнання.

У запропонованій схемі котушки газових клапанів 8, 12, 16 вмикаються послідовно із відповідними інтегральними стабілізаторами струму 9, 13, 17, які налаштовані на мінімальний струм утримання. Плюси котушок газових клапанів 8, 12, 16 підключено до відповідних виходів стабілізаторів струму 9, 13, 17, а мінуси на «масу» автомобіля 10.

У режимі роботи «бензин» на вході живлення для режиму «бензин» 2 присутній високий рівень напруги +12 В, на вході живлення для режиму «газ» 1 присутній низький рівень напруги 0 В. У режимі роботи «бензин» відбувається заряджання накопичувального конденсатора 4 до робочої напруги +12 В через розділюючий діод 3 та замкнуте комутуюче реле 5.

У режимі роботи «газ» на вхід живлення для режиму «газ» 1 подається високий рівень напруги +12 В, а на вході живлення для режиму «бензин» 2 встановлюється низький рівень напруги 0 В. У режимі роботи «газ» відбувається розряд накопичувального конденсатора 4 імпульсом напруги +12 В через контакти розімкнутого комутуючого реле 5, котушку газового клапана 8, розділюючі діоди 11, 15 та відповідні котушки газових клапанів 12, 16. Комутуюче реле 5 вмикають так, щоб воно не споживало струм утримання.

Для підвищення енергоефективності в режимі роботи «газ» котушки газових клапанів підключають до джерела живлення через інтегральні стабілізатори струму 9, 13, 17, якими індивідуально задається мінімальний струм утримання кожної котушки газового клапана 8, 12, 16. Струм утримання найкраще визначається експериментально для кожної конкретної моделі газового клапана. Оскільки, як правило, струм спрацювання газових клапанів відрізняється, у залежності від моделі клапана та його котушки, то рекомендується в ролі котушки газового клапана 8 встановити котушку газового клапана із максимальним струмом спрацювання. Наприклад, для мікросхеми стабілізатора LM317 номінал резистора R, який визначає струм стабілізації I, можна підібрати за формулою [9]:

$$I=1.25/R$$

Типова схема підключення стабілізатора струму зображена на рисунку 2.

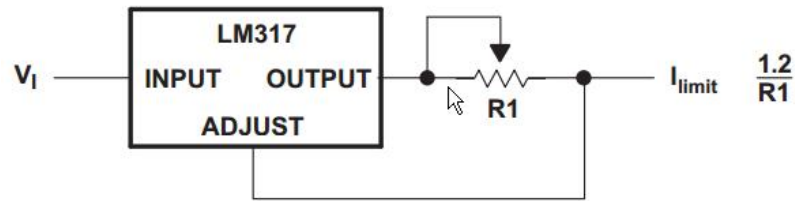


Рисунок 2 – Схема підключення стабілізатора струму [9]

Імпульси напруги від перехідних процесів у схемі гасяться шунтуючими діодами 6, 7, 14, 18. До переваг запропонованої схеми потрібно віднести високий ступінь інтегрованості в існуючі системи газобалонного обладнання без значних модифікацій та затрат на переобладнання, можливість індивідуального налаштування струму утримання для кожної котушки газового клапана, що дозволяє поєднати в одній схемі електромагнітні газові клапани із дещо відмінними параметрами котушок.

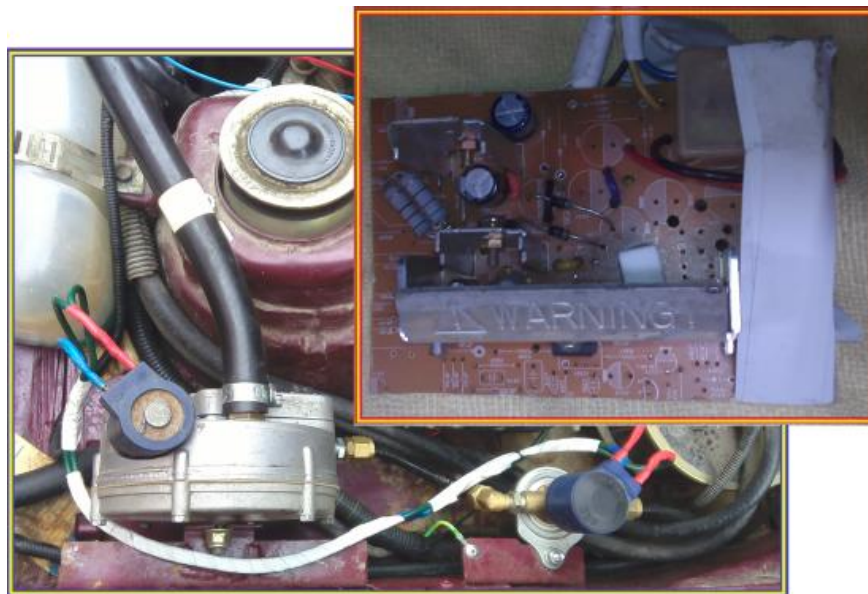


Рисунок 3 – Загальний вигляд прототипу модуля керування ЕК та система електромагнітних клапанів автомобіля.

Висновки.

Розроблений пристрій належить до області електроніки та електротехніки, зокрема, до електронного обладнання автомобілів та інших транспортних засобів, які містять у системах ГБО електромагнітні клапани, і може бути використаний для зниження енергоспоживання системи електромагнітних клапанів газобалонного обладнання автомобіля, а також для захисту бортової мережі автомобіля і підключених до неї пристроїв від імпульсів та перепадів напруги, що виникають під час їх спрацювання.

Пристрій містить накопичувальний конденсатор, розділюючі та шунтуючі діоди, який відрізняється тим, що керуючі сигнали вибору режиму роботи об'єднують із основним джерелом живлення, яке послідовно комутують або до входу живлення для режиму «бензин», або до входу живлення для режиму «газ», в залежності від виду палива, на якому працює двигун автомобіля, накопичувальний конденсатор заряджають через розділюючий діод та замкнуте комутуюче реле від джерела живлення в режимі «бензин», для підвищення енергоефективності в режимі роботи «газ» котушки газових клапанів підключають до джерела живлення через інтегральні стабілізатори струму, якими індивідуально задається мінімальний струм утримання кожної котушки газового клапана, а комутуюче реле вмикають так, щоб воно не споживало струм утримання.

Запропонований пристрій має наступні переваги:

- зниження енергоспоживання системи електромагнітних клапанів газобалонного обладнання автомобіля та похідних систем;
- високий ступінь інтегрованості в існуючі системи газобалонного обладнання без значних модифікацій та затрат на переобладнання;
- можливість індивідуального налаштування струму утримання для кожної котушки клапана, що дозволяє поєднати в одній схемі електромагнітні газові клапани із дещо відмінними параметрами котушок;
- захист бортової мережі автомобіля від імпульсів та перепадів напруги, що виникають під час спрацювання електромагнітних клапанів.

До недоліків даної схеми можна віднести те, що може бути неконтрольоване відпускання осердя клапана, що призводить до його закриття. Даний ефект спостерігається з декількох причин. Однією з них є надто занижений струм утримання, а також вібрація корпусу клапана, яка виникає внаслідок руху авто та роботи двигуна. Для усунення даної проблеми необхідно збільшити струм утримання клапана шляхом регулювання номіналу резистора у відповідному інтегральному стабілізаторі струму. Також неконтрольоване відпускання осердя клапана може виникнути і в результаті перегріву відповідного йому стабілізатора струму, який, внаслідок спрацювання теплового захисту, відключиться, розірвавши при цьому коло живлення котушки клапана. Для усунення перегріву необхідно встановити мікросхему стабілізатора на радіатор більшої площі та забезпечити краще охолодження схеми.

Однією з причин неспрацювання імпульсного запуску клапана може бути окислення, або підгоряння контактів механічного комутуючого реле 5. Тому необхідно своєчасно проводити обслуговування та заміну даного елемента. Одним із варіантів усунення даного недоліку є заміна механічного на сучасне напівпровідникове твердотільне низькоомне реле.

Також збільшення складності схеми керування клапанами ГБО у порівнянні із типовими, призводить до деякого несуттєвого зниження надійності та підвищення остаточної вартості системи в цілому.

Застосування пристрою на інжекторному авто «Славута 1,3Si» у конфігурації ГБО-2: котушка 8 (редуктор BRC Теспо 100 kW [10]) 6 Ом, котушка 12 (клапан з фільтром BRC ET98 MY07 пропан) 16 Ом, котушка 16 (мультиклапан циліндричного балона Tomasetto клас А [11]) 12 Ом, показало високу надійність запропонованої схеми, усунення збоїв двигуна під час переходів «бензин-газ-бензин», у порівнянні із стандартною схемою включення STAG2-W [12].

Отже, сферою використання даного пристрою є електронне обладнання автомобілів та інших транспортних засобів, які містять у системах газобалонного обладнання електромагнітні клапани. Сферу використання можна розширити, провівши незначні схемотехнічні зміни в пристрої.

1. И.Г. Игловский, Г.В. Владимиров. Справочник по слаботочным электрическим реле. – Л.: Энергоатомиздат, 1990.
2. Марголин Ш. М., Гуров А. С. Функциональные узлы схем автоматического управления: (справ. пособие). – М.: Энергоатомиздат, 1983. – 168 с., ил.
3. А.С. СРСР SU №1206843 А, кл. H01F7/18, 1986 р.
4. Пат. РФ RU №2180143 С1, кл. H01F7/18, 2002 р.
5. Пат. UA №110132, МПК H01F7/18, 2016 р.
6. Пат. UA №76717, МПК F02M 51/00, 2013 р.
7. Загальний курс фізики. У 3 ч. Ч. 2. Електрика і магнетизм: навч. посіб. / І.М. Кучерук, І.Т. Горбачук, П.П. Луцьк ; за заг. ред. І. Кучерука. – К. : Техніка, 2001. – 452 с.
8. Газобаллонное оборудование: Как это работает: ответы на все вопросы, возникающие при эксплуатации ГБО. Схемы установки ГБО: карбюратор / инжектор. – Монолит, 2008. – 75с.
9. <http://www.ti.com/lit/ds/symlink/lm317.pdf>
10. <https://www.brc.com.ua/>
11. <https://www.tomasetto.com/>
12. <http://www.stag.com.ua/>

УДК 681.515.8

Н.В. Здолбіцька, А.П. Здолбіцький, Б.Ю. Калінін
Луцький національний технічний університет

КОМП'ЮТЕРНА СИСТЕМА ВІЗУАЛЬНИХ ЕФЕКТІВ

Н.В. Здолбіцька, А.П. Здолбіцький, Б.Ю. Калінін. Комп'ютерна система візуальних ефектів. Розроблено програмно-апаратний комплекс для керування 6-а незалежними каналами RGB світлодіодної стрічки.

Ключові слова: RGB, світлодіод, ATmega, PyQt, ШІМ.

Рис. 5. Літ. 10

Н.В. Здолбицкая, А.П. Здолбицкий, Б.Ю. Калинин. Компьютерная система визуальных эффектов. Разработано програмно-апаратный комплекс для управления 6-ю независимыми каналами RGB светодиодной ленты.

Ключевые слова: RGB, светодиод, ATmega, PyQt, ШИМ.

N.V. Zdolbitska, A.P. Zdolbitsky, B.Yu. Kalinin. The computer system of visual effects. Developed software and hardware to control the sixth independent channels RGB LED strip.

Keywords: RGB, LED, ATmega, PyQt, PWM.

Постановка проблеми. Сучасною промисловістю виробляється велика кількість різноманітних систем “розумного” освітлення, проте переважна більшість бюджетних пристроїв виконує лише одну функцію, що призводить до нераціонального використання наявних освітлювальних пристроїв.

Актуальність. Розробка бюджетного багатофункціонального пристрою з відкритим кодом є актуальною, оскільки наразі на ринку ця ніша майже не зайнята.

Формулювання цілі. Розробити бюджетну багатофункціональну систему візуальних ефектів, яка буде реалізовувати мультимедійні функції.

Сучасні системи “розумного” освітлення. Сучасний будинок важко уявити без електричних пристроїв – від вже звичайної “економічної” освітлювальної лампи до складних комп'ютерних систем. Для спрощення керування ними та об'єднання в одну централізовану мережу розроблені системи “розумний” будинок. Це така система, яка об'єднує усі побутові пристрої для можливості простого централізованого, дистанційного чи віддаленого управління. Найпростіші електроприлади, на кшталт ламп освітлення, обігрівачів та електричні розетки, керуються простим вмиканням-вимиканням за допомогою реле [1]. Складніші пристрої, які містять в собі мікроконтролер чи мікропроцесор, отримують керуючі сигнали по одному із протоколів обміну та виконують вказану дію [2]. Найпоширеніші протоколи: RS 485, RS 232, LAN, Wi-Fi та інфрачервоний зв'язок [3].

Основними завданнями системи “розумного” будинку є керування світлом, кліматом, забезпечення безпеки, сенсорне управління, віддалене керування, голосове управління. Найпростішим в реалізації та найнеобхіднішим з точки зору частоти взаємодії користувача з системою є система освітлення.

Більшість існуючих на ринку систем інтелектуального освітлення мають або дуже обмежений функціонал [4], або дуже велику вартість [5]. Розроблений комплекс займає нішу багатофункціональних бюджетних DIY пристроїв, що забезпечується завдяки використанню платформи Arduino [6], поширеної компонентної бази, та вільного ПЗ. Ліцензування розробленого комплексу за допомогою відкритої ліцензії дозволить будь-кому додати потрібний функціонал чи вдосконалити існуючий [7].

Опис розробленої системи. Програмно-апаратний комплекс складається з таких основних блоків (рисунок 1): керуючий пристрій, ключі, джерела світла, джерело живлення, управляюча програма з 2-ма інтерфейсами (консольний та графічний). В якості керуючого пристрою було обрано плату Arduino Nano, як недорогий та поширений контролер з достатньою швидкодією. Для ключів використовуються польові N-канальні транзистори, що дозволяє підібрати необхідні параметри під джерела світла та комутувати навантаження, які живляться більшою напругою, ніж логічний рівень контролера. В якості джерела світла обрано повноколірну світлодіодну стрічку, що дає можливість встановити систему на об'єктах з будь-якою кривизною поверхні та забезпечує енергоефективність всього пристрою. Джерело живлення обирається у відповідності до параметрів джерела світла, тому на платі розташовано 2 різних роз'єми для БЖ. Arduino має вбудований стабілізатор напруги на 5 В, тому може бути заживлено від одного джерела живлення із джерелом світла.

© Здолбіцька Н.В., Здолбіцький А.П., Калінін Б.Ю.

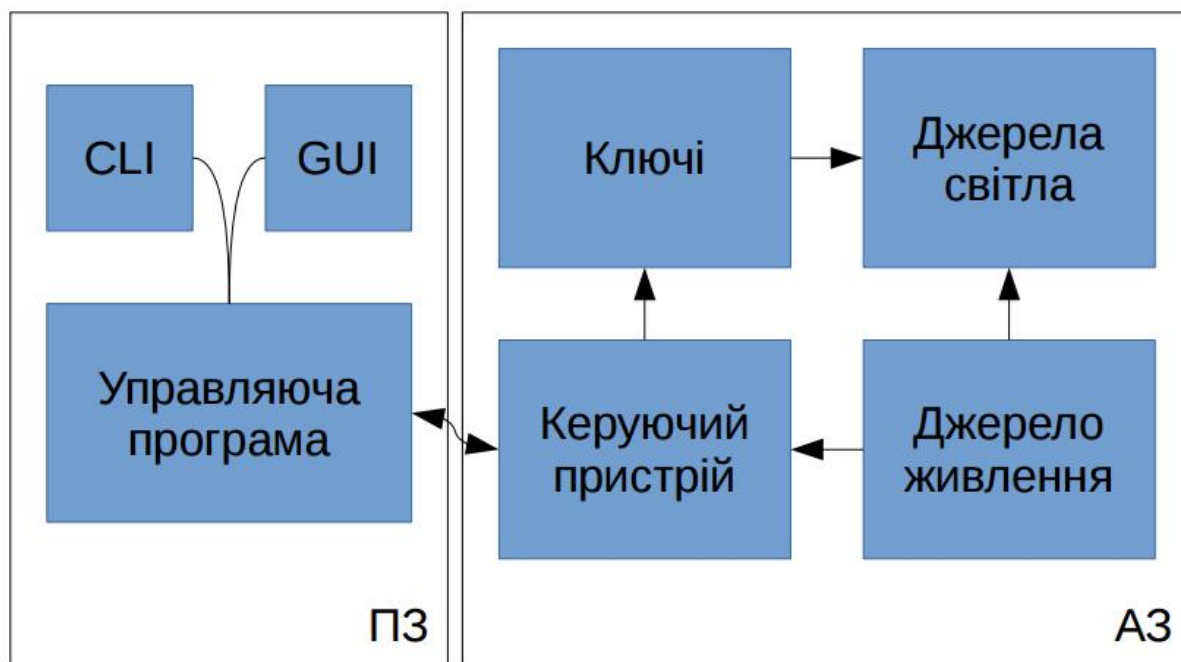


Рисунок 1 – Основні складові комплексу

Принципова схема пристрою показана на рисунку 2. ARDUINO NANO являється центральним керуючим пристроєм, він отримує живлення +12 В із зовнішнього джерела через роз'єми J1 або X7 та конденсаторний фільтр C2, C3. Через вбудовані лінійні стабілізатори Arduino віддає напругу +5 В та +3.3 В. +5 В необхідні для підтяжки кнопки U\$2 через резистор R37 та живлення інфрачервоного приймача IR1. Конденсатор C1 служить для згладжування пульсацій при замиканні чи розмиканні механічних контактів, так званий "брязкіт". Ключі на МДН-транзисторах T1...T18 керуються ШІМ сигналом із виводів Arduino A0...A5 та D2...D13. Резистори R1...R36 використовуються в якості обв'язки для транзисторів, парні – для стікання наведеного заряду із затвору на землю в простій, непарні – для захисту портів мікроконтролера від надмірного струму при виході з ладу ключа та для зменшення пікового струму переключення транзистора. Підключення світлодіодної стрічки із спільним анодом відбувається за допомогою роз'ємів J2...J7.

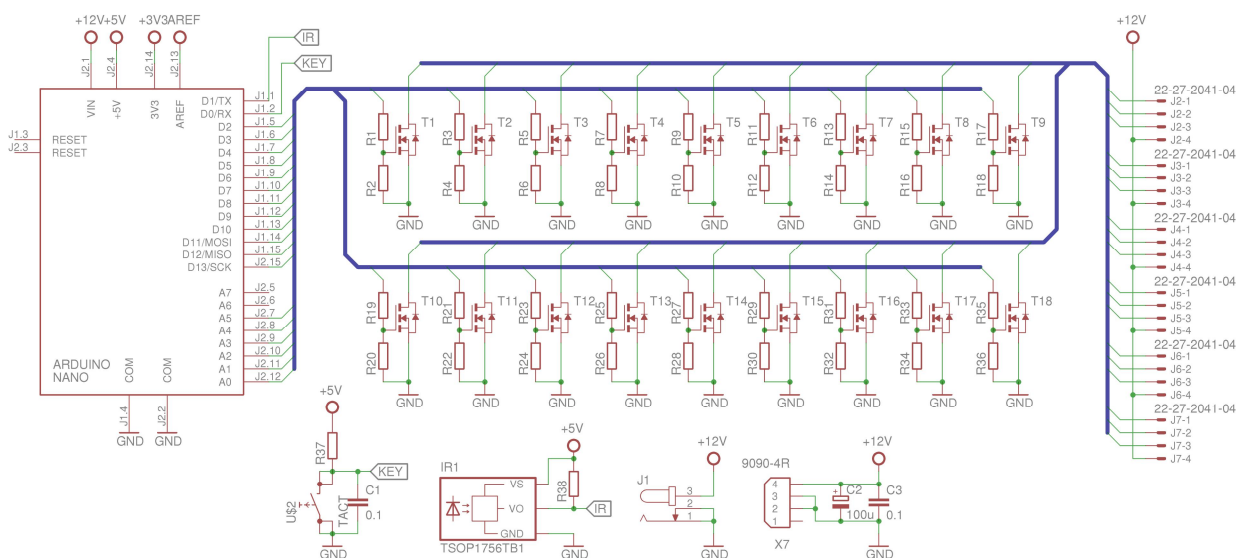


Рисунок 2 – Принципова схема пристрою

Друковану плату для пристрою було розроблено в САПР Eagle CAD (рисунок 3). Кола живлення (роз'єми та фільтри) розташовано в нижній частині плати. Від них відходять широкі

доріжки – шини +12 В та землі. По лівому і правому краю розміщено 6 роз'ємів для світлодіодної стрічки та відповідні ключі біля них. У середній частині – платформа Arduino та захисні резистори.

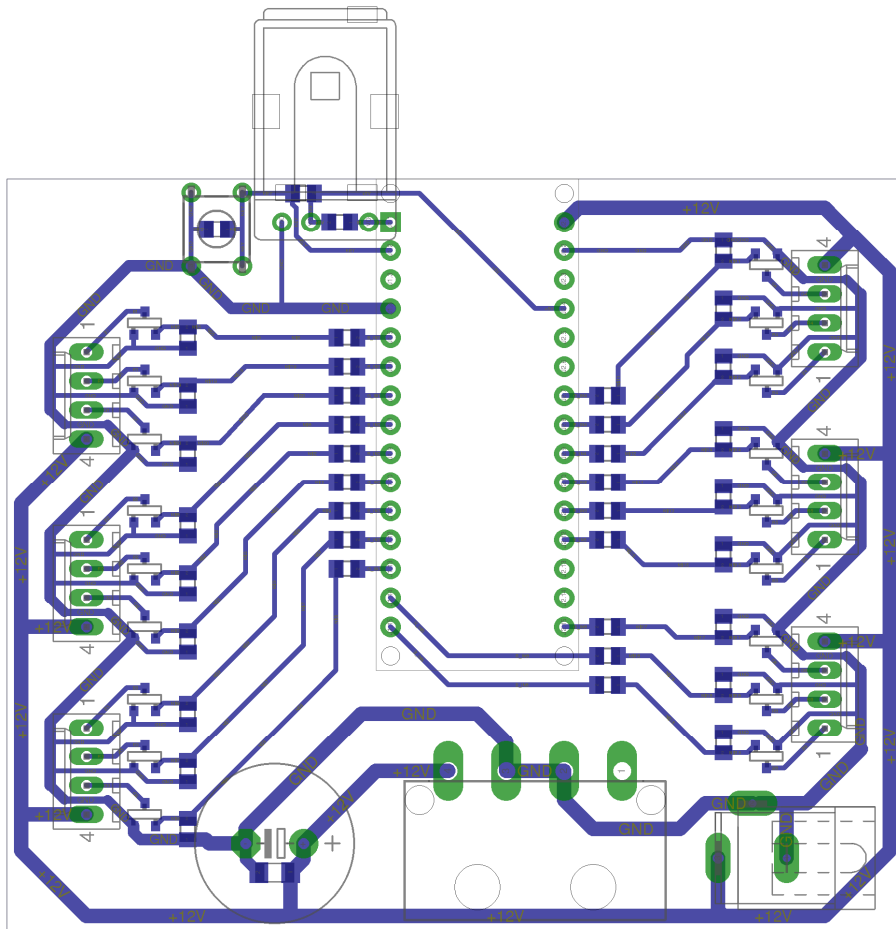


Рисунок 3 – Друкована плата пристрою

Керуюча програма для мікроконтролера написана на мові C з використанням компілятора avr-gcc та стандартної бібліотека avr-libc [8]. Структуру програми мікроконтролера показано на рисунку 4. При ініціалізації запускаються переривання від лічильника-таймера T/C0, та при отриманні чи надсиланні байта через USART0. В обробниках переривань USART0 реалізований кінцевий автомат для прийому/надсилання повідомлень через послідовний порт, тобто в програмі викликається лише функція прийому/передачі і одразу повертається управління в основну програму, а усі необхідні дії виконуються незалежно і без участі основної програми – це значно спрощує розробку і масштабування коду. На перериванні від лічильника-таймера T/C0 реалізовано програмний ШІМ на 18 незалежних каналах розрядністю 9 біт. При зменшенні кількості каналів можна збільшити розрядність ШІМ, що забезпечить плавнішу зміну кольорів на низькій яскравості.

Основна програма ж виконується у “вічному” циклі, очікуючи приходу команди по послідовному інтерфейсу. Цим забезпечується завершеність автомата світлових ефектів. Усі “інтелектуальні” функції покладені на управляючу програму на комп'ютері.

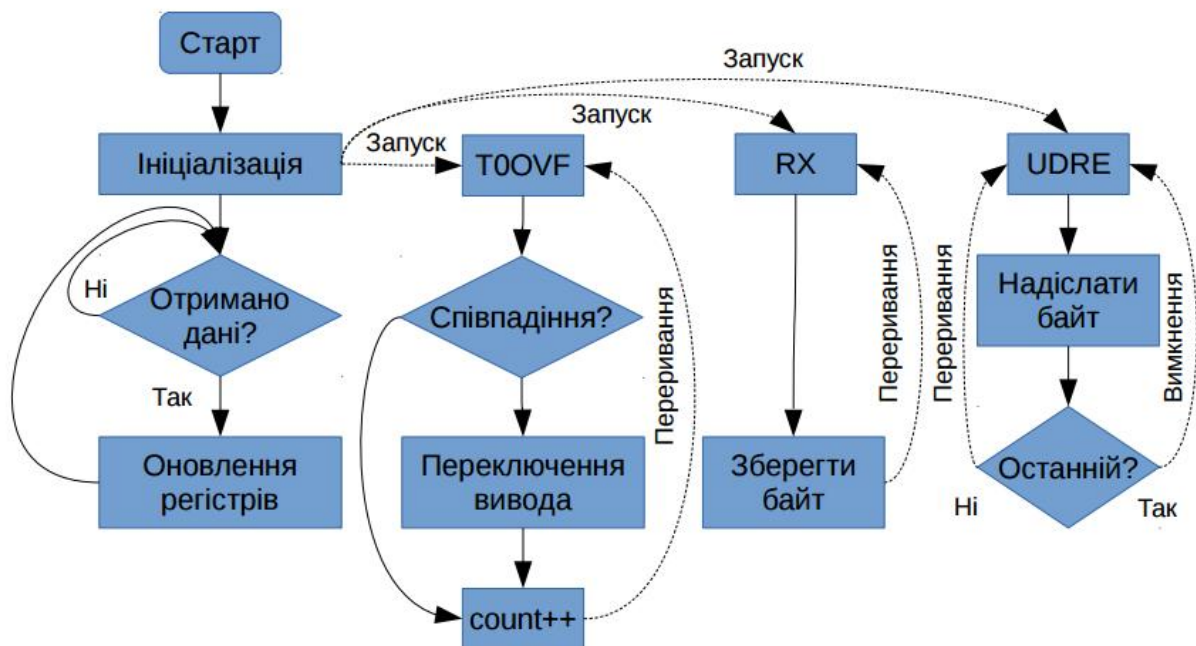


Рисунок 4 – Структура роботи мікроконтролера

Розробка управляючої програми з графічним інтерфейсом для комп'ютера проводилася на мові Python з використанням бібліотек зі стандартного репозиторію [9]. В якості графічної бібліотеки використано PyQt5, оскільки вона забезпечує кросплатформність та зручні інструменти для розробки графічного інтерфейсу. Структура програми показана на рисунку 5.

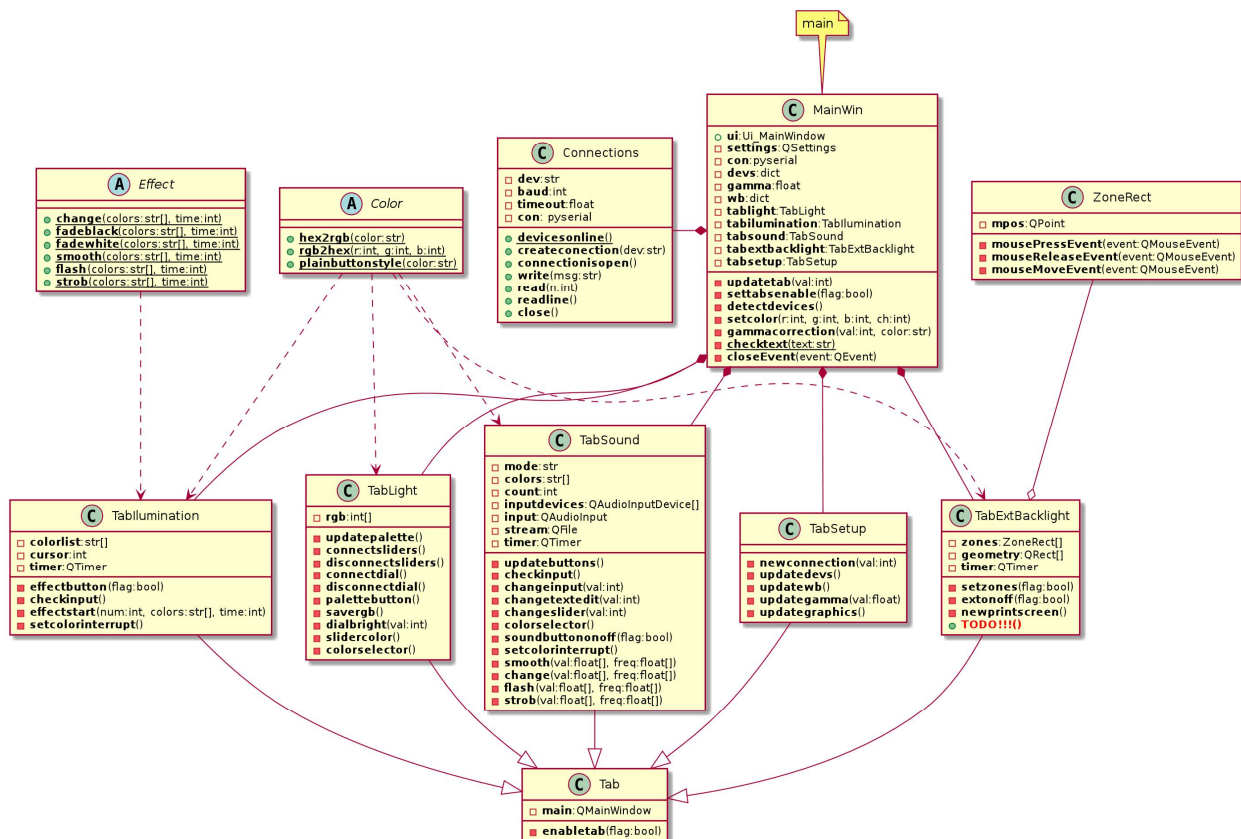


Рисунок 5 – Структура управляючої програми

Клас MainWin є головним класом програми, він керує усіма компонентами форми через об'єкт ui. Також в ньому створюються такі об'єкти як:

- settings – для збереження та відновлення усіх налаштувань та параметрів вікна;
- con – для обміну даними через послідовний порт;
- tablight, tabillumination, tabsound, tabextbacklight, tabsetup – для взаємодії з відповідними вкладками головного вікна.

Клас Tab є батьківським для усіх інших Tab*-класів.

У класі TabLight реалізовано інтерфейс взаємодії для реалізації функцій простого світильника. Він дозволяє обрати колір з палітри або ввести RGB-значення та відрегулювати яскравість.

Клас TabIllumination реалізує динамічну зміну кольорів за наперед заданим користувачем законом. Реалізовані різні варіанти переходів: просте переключення, перехід через чорне, вицвітання до білого, плавна зміна кольору, ефект спалаху та ефект стробоскопу. На вкладці реалізовано 4 секції для зберігання різних наборів кольорів та переходів між ними.

У класі TabSound реалізовано можливість вибору вхідного каналу звукових даних із усіх наявних в системі пристроїв, вибір кольору для низьких, середніх та високих частот, їхні смуги, рівень шуму та регулювання загальної яскравості. Реалізовані 4 режими роботи: звичайний – кількість кольору залежить від амплітуд частот, зміни – програма реагує на низькочастотні біти та змінює колір на наступний на кожному з них та режими спалаху і стробоскопу – реакція на біти із одинарним або подвійним спалахом на них.

Клас TabExtBacklight реалізує функціонал схожий до технології Ambilight фірми Philips[10]. Суть її полягає в аналізі зображення на екрані і підсвіченні зон позаду екрану відповідним кольором, що збільшує занурення в мультимедійний контент. На вкладці реалізовано можливість вибору кількості зон, інтервалу оновлення та вибору розміру і розташування зон на екрані.

У класі TabSetup реалізовано вибір доступних пристроїв, налаштування балансу білого та гама-корекції.

Клас ZoneRect реалізує прозорі прямокутні вікна зі зміною розмірів для вибору зон задньої підсвітки. Клас Connection реалізує інтерфейс взаємодії із послідовним портом. У класі Color реалізовані функції для перетворення хеш-коду кольору в значення R, G, B та навпаки і, також, генерації стилю кнопки в залежності від її кольору. Клас Effects реалізує генерацію таблиць переходів кольорів з одного в інший в залежності від різних ефектів.

Висновки. Здійснено реалізацію програмно-апаратного комплексу на базі платформи Arduino Nano (ATmega162). Проект повністю розроблений “з нуля” з використанням відкритого інструментарію. У системі реалізовано режим освітлення, ефектів, світломузики та фонові підсвітки зображення.

1. Gerhart James Home Automation and Wiring//McGraw-Hill Professional, 31 March 1999. – ISBN 0-07-024674-2.
2. Harper et al, 2003. pp. 18–19
3. Mann William C. The state of the science//Smart technology for aging, disability and independence. – John Wiley and Sons, 7 July 2005. – ISBN 0-471-69694-3.
4. <http://www.theverge.com/2016/8/23/12560024/best-smart-lights-home-gadgets-philips-hue>
5. <https://www.control4.com/solutions/smart-lighting>
6. <https://www.arduino.cc/en/Main/ReleaseNotes>
7. Richard Stallman. Why «Open Source» misses the point of Free Software
8. Артур Гриффитс. GCC. Настольная книга пользователей, программистов и системных администраторов. – Диасофт, 2004. – С. 624. – ISBN 966-7992-34-9.
9. http://docs.linux.org.ua/Програмування/Python/Підручник_мови_Python/
10. <http://www.philips.ru/c-m-so/televisions/p/ambilight>

УДК 622.232.8

Каганюк О.К.

Луцький національний технічний університет

АНАЛІЗ СИСТЕМ АВТОМАТИЧНОГО КЕРУВАННЯ ВУГІЛЬНО - ДОБУВНИМ КОМБАЙНОМ В СКЛАДНИХ ГІРНИЧО-ГЕОЛОГІЧНИХ УМОВАХ.

Каганюк О.К. Аналіз систем автоматичного керування вугільно - добувним комбайном в складних гірничо-геологічних умовах. В даній статті проводиться аналіз та будова систем автоматичного управління вугільно-добувними комбайнами для можливої роботи в складних гірничо-геологічних умовах.

Ключові слова: вугільно-добувний комбайн, система автоматичного управління, одноконтурна система, двоконтурна система.

Каганюк А.К. Анализ систем автоматического управления угледобывающим комбайном работающего в сложных горно-геологических условиях. В Данной статье проводится анализ и построение систем автоматического управления угледобывающими комбайнами, работающими в сложных горно-геологических условиях.

Ключевые слова: угледобывающий комбайн, система автоматического управления, одноконтурная система, двухконтурная система.

Kaganiuk A.K. Analysis of automatic control systems for coal mining combine operating in difficult mining and geological conditions. This article analyzes and constructs automatic control systems for coal mining combines operating in complex mining and geological conditions

Keywords: coal mined processor, automatic control system, flue system, dual-circuit system.

Підвищення продуктивності праці і інтенсифікації виробництва в вугільній промисловості є найбільш **актуальна** в сучасній промисловості по автоматизація технологічних процесів, а також впровадження промислових роботів і маніпуляторів.

Сучасні вимоги до систем автоматичного керування забійним обладнанням припускають рішення **проблеми** за рахунок інтенсивного і високопродуктивного видобутку вугілля без постійної присутності людей в забої.

Відомі дослідження і розробки систем управління вугледобувними комбайнами в профілі пласту дозволяють розділити їх на два основних класи: системи автоматичного управління в профілі пласта (далі САУ ПП) і системи програмного управління в профілі пласта (далі СПУ ПП). Системи автоматичного управління в профілі пласту здійснюють управління комбайном в профілі пласта безпосередньо на основі інформації датчика «порода вугілля» (далі ДПВ), яка поступає при переміщенні комбайну по довжині лави. САУ ПП при відповідному налаштуванні, беруть на себе всі повноваження по управлінню ріжучими органами вугільно – добувного комбайна без втручання людини.

Системи програмного управління в профілі пласту СПУ ПП припускають дистанційне управління ріжучим органом по програмі, яка записується в програмний блок на основі даних огляду лави через декілька циклів роботи. Більш раціональним рішенням є реалізація автоматичного програмного управління, яка припускає автоматичний збір інформації про товщину вугільної пачки по довжині лави, що дає можливість виключити трудомістку ручну операцію по збору даних про товщину вугільної пачки, яка остається як в «почві» так і в «кровлі» пласта, при знятті вугільної стружки. [15]. При такому варіанті збору інформації, необхідно розташовувати датчики контролю по всій довжині лави. При такому аналізі, виникає задача оптимального розташування тих датчиків контролю по довжині лави з раціональним шагом на відповідній відстані останніх між собою, бо від цього буде залежати точність відпрацювання СПУ ПП. Але це є **проблематичним** питанням.

По точності і ефективності управління САУ ПП випереджують СПУ ПП, оскільки в САУ ПП може бути збудовані, як слідкуюча система, яка дозволяє оперативно поставити інформацію в блок управління без всякого транспортного спізнання. Крім того, будова САУ ПП дозволяє реалізувати її в вигляді бортових регуляторів, що виключає необхідність використання ліній зв'язку і систем телемеханіки і дозволяє практично ідеально відслідковувати границю розділу між породою та вугіллям. Це дуже важливий аргумент при роботі вугільно – добувного комбайна в складних гірничо – геологічних порушень (далі ГПП)

Однак, така система не дозволяє використати всі свої можливості, коли в лаві виникають дуже складні геологічні умови, які не підлягають технічному опису. В цьому випадку необхідно доповнювати дану систему додатковим блоком програмного управління, якій дозволяє проходити ці

складні геологічні порушення з раціональною технологічною схемою видобуду вугілля, притому область функціонування програми повинна задаватись за допомогою датчику місцезнаходження комбайну (далі ДМК). Але в цьому випадку САУ ПП спроможна виконувати визначені функції контролю формованої машинної дороги і фіксувати факт проходження ГПП. В цьому зв'язку ідеальним представляється поєднання САУ ПП з елементами програмного управління. Способи побудови таких САУ ПП обумовлені рядом наступних основних факторів:

- наявність і тип використовуваних датчиків;
- спосіб і конструктивні можливості управління ріжучими органами;
- можливість і спосіб управління кутом нахилу комбайну;
- вимоги до точності регулювання і якості формованої машинної дороги.

Будова такої системи визначається вибраним способом управління комбайном в профілі пласту [12,]. Можливі варіанти цих способів ілюструються класифікаційною схемою на рис.1.1.



Рис. 1.1. Класифікаційна схема способів управління комбайнами в профілі пласта

Відмітимо зразу, що по результатам обробки керуючих впливів нахил комбайну відносно поперечної осі, еквівалентний плоско-паралельному переміщенню ріжучих органів, а з точки зору простоти і точності управління уступає йому.

Нахил комбайну відносно поздовжньої осі, застосований для управління в профілі пласту автономно, обмежений допустимим кутами взаємного скручування рештаків конвеєру і при неспокійній гіпсометрії не здатен забезпечити необхідний діапазон регулювання. Найбільш ефективним способом управління в профіль пласту являється двоконтурне управління, при якому поєднується плоско-паралельне переміщення ріжучого органа (далі РО) і нахил комбайну відносно поздовжньої осі.

Нижче наведений аналіз результатів практичної реалізації способів управління.

. Обов'язковою умовою створення такої САУ ПП являється ДПВ. Як відмічається в літературних джерелах [17,18,19]. існують ДПВ відкритої і прихованої границі «порода-вугілля». Перший тип ДПВ видає інформацію практично релейного типу, фіксуючи факти переходу РО із вугілля в породу і навпаки. Відповідно, САУ ПП, в який використовується ДПВ відкритої границі «порода-вугілля», будуються в більшості як релейна двопозиційна система управління [17,18,19,20]. Характерна структурна схема таких САУ ПП приведена на рис.1.2.

Така САУ ПП була реалізована на комбайні 2К-52 (сумісна розробка ІГД ім. А.А.Скочинського і Гіпровуглемашу) [8]. Те, що дана система не дала позитивних результатів при проведенні її випробувань, пояснюється цілком і повністю недоліками при її побудові і технічній реалізації. Як видно з рис.1.3, ДПВ знаходився на «старій» машинній дорозі, що і обумовило запізнення на цикл $\tau_{\text{ц}}$ (стружку) по ступання інформації. Випадковий характер мінливості гіпсометрії пласта і наявність штибу обумовили велику похибку регулювання. Система працювала нестійко, крім того, використання ДПВ притискного типу викликало більше складностей при його прив'язці і малу надійність його функціонування. Будучи притиснути до ґрунту пласту гідродомкратом, датчик діяв як якір, спиняючий подачу комбайну.

Принципово нові можливості в створенні САУ ПП дала розробка безконтактного ДПВ, вбудованого безпосередньо в ріжучий органу комбайна [10]. На основі такого ДПВ був розроблений і успішно випробуваний цифровий регулятор для управління комбайном (1ГШ-68) в профілі пласта [11]. Структурна схема такої САУ ПП приведена на рис.1.4.

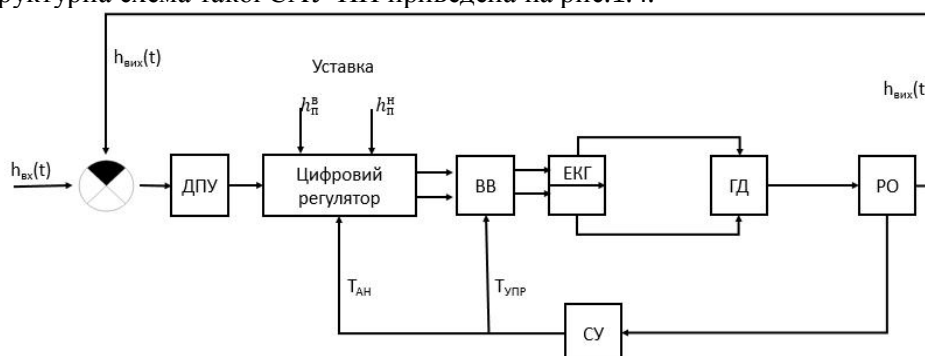


Рис. 1.4. Структурна схема САУ ПП «Квант» на комбайні 1ГШ-68

Сигнал з безконтактного ДПУ, розміщеного в зоні кутових зубків РО, надходить в цифровий регулятор, створений на базі двохпорогового частотного компаратора [6]. Накопичення сигналу в цифровій формі проводиться протягом часу ($T_{\text{ан}}$), що задається внутрішнім лічильником, на який, в свою чергу, надходять імпульси синхронізуючого пристрою (далі СП). СП сприймає імпульси з кожним оборотом РО, тобто з частотою $= 1\text{Гц}$. Сигнал з ДПВ, інтегрований за час ($T_{\text{ан}}$) порівнюється з цифровими уставками ($H_{\text{пв}}$) і ($H_{\text{пн}}$), в результаті чого формується дво полярний сигнал управління, тривалість якого визначається елементом витримки часу. Цей елемент також синхронізується імпульсами СП. Керуючі впливи включають електрогідроклапани (далі ЕГК) і пускають у хід гідродомкрат (далі ГД), що переміщає поворотний редуктор з РО. Дана САУ ПП – швидкодіюча слідкуюча система без транспортного запізнювання, яка здатна з високою точністю відстежувати лінію розділу «порода-вугілля», залишаючи в процесі регулювання запобіжну вугільну пачку заданої товщини. Однак ретельний аналіз дозволяє встановити ряд принципових обмежень області її функціонування, що робить малоперспективною її практичне використання. Основний недолік даної САУ ПП, що є наслідком її точності і швидкодії, полягає в тому, що вона відстежує лінію розділу «порода-вугілля» при будь-якій її зміні, прагнучі точно її повторити. При цьому формується машинна дорога з усіма впадинами і виступами, наявними на лінії розділу «порода-вугілля». У той же час відомо, що забійний конвеєр, в силу обмеження рухливості риштаків, накладає суворі обмеження на висоту уступів і абсолютні перепади висот на довжині риштака. Так, кожен окремий риштак не може бути поміщений на уступ вище 3 см, а при великому перепаді висот на краях риштака кут його вигину щодо сусіднього може перевищити допустимі (3гр), що призведе до поломки конвеєра. Таким чином, в цій системі не вистачає блоку обмежень відпрацювань, що безумовно знижує точність регулювання, але гарантує надійне функціонування комплексу.

Другий недолік системи пов'язаний з принципом її синхронізації. При змінній в широких межах швидкості подачі комбайна (від 0 до 10 м/хв) сталість часу вимірювання призводить до формування майданчиків змінної довжини, що створює додаткові труднощі при укладанні конвеєра на сформовану дорогу. Нарешті, головний недолік пов'язаний з тим, що за принципом управління дана система є одно контурною, оскільки впливає лише на вертикальне положення РО. У той же час показано, в тому числі і дослідженнями автора при проведенні випробувань, що переміщення комплексу в напрямку простягання пласта пов'язано з подштибовкою забійного конвеєра [14]. При

цьому в автоматичному режимі управління має бути ручна операція видалення штибу з машинної дороги.

В результаті буде мати місце систематичне «спливання» комплексу, що приводить до зменшення діапазону переміщень вниз ріжучого органу. Через кілька відрізів (в залежності від темпу підштибовки) [12].цей діапазон може бути вичерпаний, що зробить САУ ПП неідеальною. Те ж явище неминуче буде спостерігатися при природній зміні кута залягання пласта, відстежити які шляхом управління плоско-паралельним переміщенням РО можна лише у виняткових випадках малих його змін. При цьому слід зазначити, що одночасні підштибовка конвеєра і зміна кута залягання в сторону почви пласта погіршують ситуацію. Дія даних факторів вивчалася і описана в ряді робіт [23,]. Випробування комплексів .Ролфа показали, що системи регулювання, що виконують тільки функції стабілізації товщини вугільної пачки, не дозволяють вирішити задачу управління рухом комплексу в профілі пласта, оскільки в даній ситуації не враховуються деякі фактори, що впливають на стан конвеєра щодо почви пласта, а також особливості останнього при його пересуванні на нову машинну дорогу. Це, в свою чергу, призводило до нестабільного стану вугледобувної машини на конвеєрі при виїмці чергової стружки і виникненню нових спотворень профілю забою. При цьому уступи, утворені на почві пласта погіршували не тільки пересувку забійного конвеєра, а й приводили до деформації окремих секцій конвеєра [2, 20,]. Це призвело до необхідності в подальших шахтних експериментах враховувати кутове положення конвеєра по простяганню пласта і здійснювати управління кутом нахилу комбайна щодо його поздовжньої осі.

При такому способі управління на почві пласта утворюються незначні по висоті сходинок, що не перешкоджають пересуванню забійного конвеєра на нову машинну дорогу.

У фундаментальному дослідженні [23] дано обґрунтовані аргументи щодо доцільності розробки системи, яка поєднувала в собі обидва способи управління - плоско-паралельним переміщенням РО і кутом нахилу комбайна. Однак до теперішнього часу відсутній позитивний досвід створення таких систем. Основна причина відсутності вітчизняних двоконтурних САУ ПП пов'язана з тим, що до теперішнього часу серійно випускаємо вугільно – добувні комбайні і не оснащуються електрогідроперемикачами гілки гідравлічного управління кутом нахилу комбайна. Друга причина пов'язана з відсутністю датчиків кута нахилу, придатних для роботи в шахтарських умовах в складі бортової автоматики. Слід зазначити, що, якщо ставиться завдання підтримки кута нахилу, відповідного кутку залягання пласта, існує можливість створення двоконтурної системи без датчиків кута нахилу. Відомий закордонний досвід створення системи управління кутом нахилу не може бути однозначно використаний на вітчизняних комбайнах. У двоконтурній системі на комбайні Андертон (шахта Ньюстед, Великобританія) зміна кута нахилу здійснювалося за допомогою спеціальної платформи [21], що не застосовується в конструкції вітчизняних комбайнів. Проте, позитивні результати даних випробувань показали, що при управлінні кутом нахилу формується плавна траєкторія укладання конвеєра і на машинній дорозі відсутні сходинок, що перешкоджають засувці конвеєра.

Висновки:

Аналіз систем автоматичного управління вугільно – добувними комбайнами в профільній площині вугільного пласта дозволяє зробити наступні допущення щодо створення систем управління:

1. Найбільш ефективним для реалізації в системах автоматичного управління вугільно – добувними комбайнами є засоби контролю скритої границі розділу між породою та вугіллям які створені на основі радіаційних методів контролю товщини вугільної пачки.
2. Для досягнення високої якості управління вугільно – добувним комбайном в профілі пласта може бути прийнята до розробки двоконтурна система управління , яка включає в собі плоско – паралельне управління РО та зміною кута нахилу комбайном.

1. . Автоматический контроль положения режущего барабана угольных комбайнов. – Mining Journal 1976,286, № 7326, p.59 – 60.
2. Автоматическое управление комбайнами на шахте Уолстентон: Перевод № 1295/71. М.: Гипроуглемаш, 1971г. 33 с.
3. . Автоматическое управление режущими органами угледобывающих комбайнов. Черняк З.А. и др. – «Безопасность труда в промышленности», 1982г, №15, с. 32-33.
4. Вебб Р. Автоматическое управление в профиле пласта комбайном Андертон. The Mining Engineer, 1967, V/ 196,№80. P.397.

5. Датчик «порода – уголь», Ильюша А.В., Солнцев В.Б. М., 1981. – 5 с. – Рукопись представлена ИГД им. А.А. Скочинского. Деп. В ЦНИЭИуголь 28 дек. 1981г. № 2242 уп – Д81.
6. А.С. № 790264 (СССР), Двухпороговый частотный компаратор/ ИГД им. А.А.Скочинского, и Научно-производственное объединение «Автоматгормаш»; Авт. изобретения Каганюк А.К. и др. Заявлено 08.06.78. № 2624593 Приоритет изобретения от 08.06. 1978г. Зарегистрировано в Государственном реестре изобретений СССР от 21. 08. 1980г.
7. Мелькумов Л.Г. , Петунин Н.С. Автоматизация угледобывающих комбайнов: Обзор – М., 1976г. – 50 с.
8. Промышленное опробование авторегулятора ИДА – 3 на комбайне 2К – 52 /Черняк З.А. и др. – В кн.: ИГД им. А.А. Скочинского. Науч. Сообщ. Вып. 69. М., 1969г, с.22 – 28.
9. Разработка методов оптимизации параметров очистных комбайнов в системе управления: Экспериментально – исследовательские работы по программному управлению угольными комбайнами в плоскости пласта: Отчет /Автоматгормашу Научный руководитель работы к.т.н. Ярыгин Б.Е. Инв.№ Б 560527. – Донецк, 1976г. – 157с.
10. Регулятор с изотопным датчиком «порода – уголь» для управления очистными комбайнами по гипсометрии пласта. /Черняк З.А., Нунупаров Г.М. , Омельченко Н.П. – Уголь, 1975, №12, с. 38 – 39.
11. А.С. №1270321 (СССР) Система автоматического управления комбайном в профиле пласта. / «НПО Автоматгормаш»: Каганюк А.К., Евстафьев Д.И. и др. Приоритет изобретения 01.04.85г. № 3875725; Зарегистрировано в Государственном реестре изобретений СССР 15.07 1986г.
12. Рудановский А.А. О рациональных способах автоматического управления угледобывающими комплексами в профильной плоскости пласта . – Уголь Украины, 1969г., №7, с. 19 – 21.
13. А.С. № 1270323 (СССР) Способ автоматического управления очистным комбайном по гипсометрии пласта и система для его осуществления Каганюк А.К., Евстафьев Д.И. и др. Приоритет изобретения 22.04.85г. № 3908263; Зарегистрировано в Государственном реестре изобретений СССР 15.07 1986г.
14. Рудановский А.А. Черников И.В. Статистические исследования подштыбовки забойных конвейеров. - В кн.: ИГД им. А.А. Скочинского. Науч. Сообщ. Вып. 84. М., 1971г., с. 105-110.
15. Датчик «порода-уголь». Ильюша А.В. Солнцев В.Б. М., 1981.-5с. Рукопись представлена ИГД им. А.А.Скочинского. Деп. в ЦНИЭИуголь 28 дек. 1981г., №2242 уп-Д81..
16. А.С. № 1250651 (СССР).Способ програмного управления угледобывающим комбайном и система его осуществления. / «НПО Автоматгормаш»: Каганюк А.К., Фрегер Д.И. и др. Приоритет изобретения 28.12.84г. № 3831914; Зарегистрировано в Государственном реестре изобретений СССР 15.04 1986г.
17. Хенкель Э., Хек Д. Регулирование горизонта резания комбайнов с барабанным исполнительным органом. _ Глюкауф. 1977г., №23., с. 10 – 15
18. Ярыгин Б.Э. Автоматическое и дистанционное управление горными машинами, комплексами и агрегатами. Киев: Знание, 1977г.- с.12.
19. Ярыгин Б.Э. и др. Система автоматического управления исполнительными органами комбайна с электрогидравлическим датчиком «порода – уголь» типа «Рубин». – Шахтная автоматика. Киев: Техника, 1975г. – 47с.
20. Cope H/R/ Philosophy and future developments of ROLF. – The Mining Engineering1965g/ V/125, №62.
21. Патент. 1.187.374(Анг.). Improvements in underframe for cool cutting machnes/ L.I. Madden. Заявлено 08.04.70.кл. E21с. .
22. Плюснин М.И., Мараев И.Д. К обоснованию индукционного способа контроля движения угледобывающих машин по гипсометрии пласта (антрациты). Изв. ВУЗов СССР. Геология и разведка, 1980, №9, с. 119 – 124.
23. Рудановский А.А. Основы автоматического управления комбайнами в профиле пласта: автореферат док. диссертации. / ИГД им. А.А. Скочинского. М.- 1972. 41 с.

УДК 004.4(06)

Коцюба А.Ю., Мацібора А.С.

Луцький національний технічний університет

ІНФОРМАЦІЙНА СИСТЕМА АНАЛІЗУ ПОСТАЧАЛЬНИКІВ СИРОВИНИ НА ПІДПРИЄМСТВО

Коцюба А.Ю., Мацібора А.С. Інформаційна система аналізу постачальників сировини на підприємство. На базі середовища Embarcadero C++ Builder XE та СКБД MySQL розроблено комплекс програм (інформаційну систему), який дозволяє проводити аналіз статистичних даних, одержаних від постачальників сировини на підприємство. За допомогою ADO-технології налагоджено взаємодію додатка із базою даних. Отримана інформаційна система дає можливість слідкувати за поставками сировини, добавляти нових (або видаляти) постачальників та виводити звітну інформацію, наприклад, про рейтинг постачальників у вигляді графіку. Рейтинг будується на основі кількості якісної продукції та її ціни.

Ключові слова: інформаційна система, автоматизація, C++ Builder проект, ADO-технологія, СКБД MySQL.

Коцюба А.Ю., Мацібора А.С. Информационная система анализа поставщиков сырья на предприятие. На базе среды Embarcadero C++ Builder XE и СУБД MySQL разработан комплекс программ (информационную систему), который позволяет проводить анализ статистических данных, полученных от поставщиков сырья на предприятие. С помощью ADO-технологии налажено взаимодействие программы с базой данных. Полученная информационная система дает возможность следить за поставками сырья, добавлять новых (или удалять) поставщиков и выводить отчетную информацию, например, о рейтинге поставщиков в виде графика. Рейтинг строится на основе количества качественной продукции и ее цены.

Ключевые слова: информационная система, автоматизация, C++ Builder проект, ADO-технология, СУБД MySQL.

Kotsyuba A.Yu, Matsibora A.S. Information system for analysis of raw material suppliers to the enterprise. Based on Embarcadero C++ Builder XE and DBMS MySQL, a set of programs (information system) has been developed that allows analysis of statistical data received from raw material suppliers to the enterprise. Using ADO-technology, the program interacts with the database. The received information system makes it possible to monitor the supply of raw materials, add new (or delete) suppliers and output reporting information, for example, on the rating of suppliers in the form of a graph. The rating is based on the quantity of quality products and its price.

Keywords: Information system, automation, C++ Builder project, ADO-technology, DBMS MySQL.

Актуальність та постановка проблеми. Підприємство – самостійний суб'єкт господарювання, зареєстрований компетентним органом державної влади або органом місцевого самоврядування, для задоволення суспільних та особистих потреб шляхом систематичного здійснення виробничої, науково-дослідної, торговельної, іншої господарської діяльності в порядку, передбаченому Господарським кодексом України та іншими законами. Існує багато підприємств, які виготовляють продукт для кінцевого споживача чи переробляють сировину для подальшого її використання у виробництві. Одним з аспектів успіху такого бізнесу є надійні постачальники сировини, які можуть запропонувати якісну продукцію за вигідною ціною.

В зв'язку з розвитком комп'ютерних технологій, багато підприємств цікавляться автоматизацією виробництва та процесів закупівлі і продажу. На програмному ринку постало завдання виготовлення не дорогого, якісного і зручного у користуванні програмного забезпечення.

Система аналізу постачальників сировини дозволяє спростити організацію виробництва і контроль роботи з постачальниками, забезпечити підприємство якісною сировиною, знизити витрати на ресурси та підвищити ефективність бізнесу. Тому підприємці та організації прагнуть забезпечити своє виробництво якісною, недорогою та простою у використанні програмою аналізу постачальників сировини.

Створення такої програми пов'язане з використанням великих об'ємів інформації та зв'язку з базами даних. Тому потрібно забезпечити захист інформації, щоб запобігти її витоку та несанкціонованого доступу до неї. Додатки такого типу створюються за допомогою ADO-технології, яка дозволяє представляти дані в об'єктно-орієнтованому вигляді [1].

Розроблена програма дозволяє контролювати роботу з постачальниками, вибирати з них кращих та ефективно розвивати свій бізнес.

Аналіз останніх досліджень та публікацій. Більшість програмних комплексів (зроблених на замовлення під конкретні потреби підприємства), що дають змогу керувати та проводити аналіз поставок сировини не є вільно доступними. Це пов'язано з тим, що такі програми є комерційною таємницею. Проте деякі компанії надають опис таких продуктів на своїх web-сторінках. Звичайно тут не маються на увазі загальновідомі пакети програм, які зазвичай здатні вирішувати всі поставле-

ні проблеми (але їх використання вимагає значної компетенції, і адаптація під незначну специфіку роботи підприємства пов'язана зі значними труднощами), наприклад, "1С: Підприємство" тощо.

Наприклад компанія "УСУ" надає вичерпну інформацію про свою програму управління виробництвом та детальну інструкцію до неї (також дозволяє встановити пробну версію [2]). Подібне рішення пропонує компанія "Фокс Менеджер": їхній продукт "Fox Manager 2.5 BPA" дозволяє вести контроль діяльності підприємства, виводити аналіз бізнес процесів [3]. Дані продукти надають можливість контролювати роботу з постачальниками, але вони дорогі і не підходять для малих підприємств, також ними складно користуватись.

Доволі цікавою на нашу думку є програма "PriceMatrix". Однією із основних функцій даної інформаційної системи є визначення кращої ціни продукції серед постачальників [4]. Хоча дана програма виконує поставлене вище завдання, але нажаль вона призначена лише для магазинів і більшість функцій нам зовсім не потрібна. В ній інтерфейс реалізовано на англійській мові, що ускладнює користування.

Мета роботи – створення інформаційної системи для підприємств, яка могла б полегшити співпрацю з постачальниками сировини в напрямку оптимізації співвідношення "ціна ~ якість ~ кількість".

Виклад основного матеріалу дослідження та обґрунтування отриманих результатів. Кожне підприємство займається своїм видом діяльності. Деякі з них досить часто виготовляють власну продукцію на основі обробки вхідної сировини; якість, ціна та кількість якої залежить від постачальника. Власне на таке підприємство і орієнтована інформаційна система, розроблена в даній роботі. Очевидно, що різновид сировини залежить від його роду діяльності підприємства. Нехай основним видом вхідної сировини в нашому підприємстві будуть свині призначені для забою. Тоді власною продукцією компанії можуть бути м'ясні вироби із свинини. Немає сенсу розповідати про важливість відслідковування якості сировини та того, щоб м'ясопереробний комбінат не виявився в кінцевому випадку збитковим. Розроблена програма використовується на етапі прийняття сировини та після першого етапу обробки свинини. Головне вікно програми показано на рис. 1.

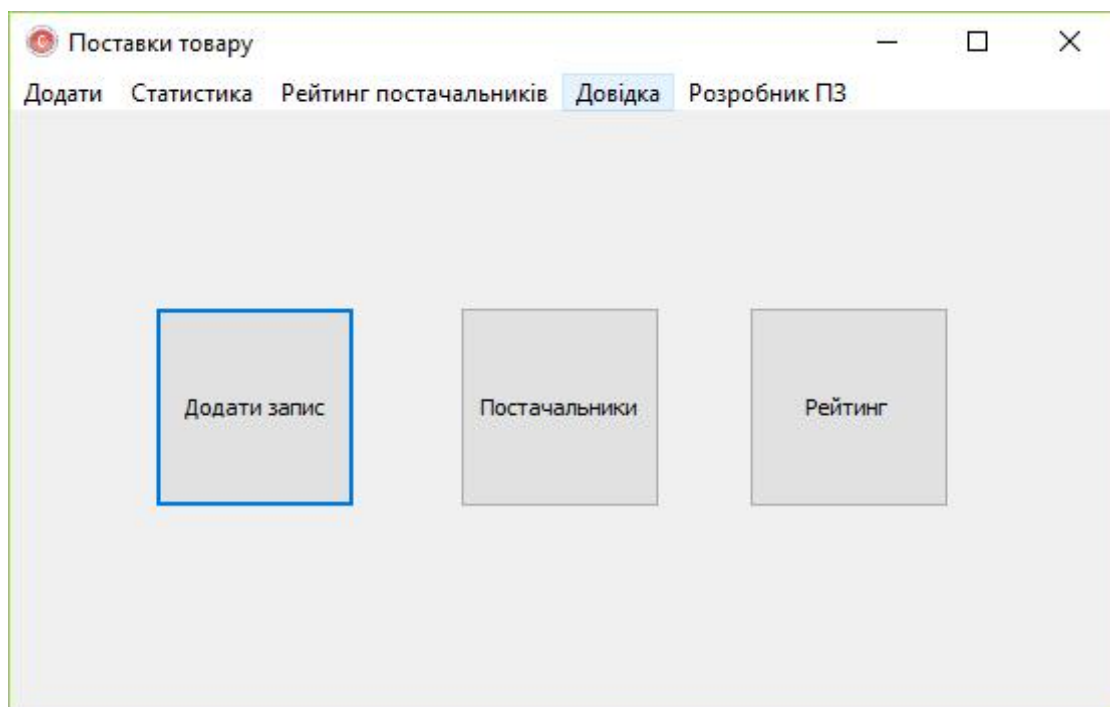


Рис. 1. Головне вікно C++ Builder проекту

Прийняття партії сировини здійснюється за наступною схемою:

- Спочатку свиней призначених до забою перевіряє ветеринар, звіряючи при цьому всі сертифікати.
- Далі свині зважуються. Сумарна маса партії, її номер, ціна та кількість голів у партії заносять в базу даних за допомогою розробленого у програмі першого діалогового вікна (рис. 2).

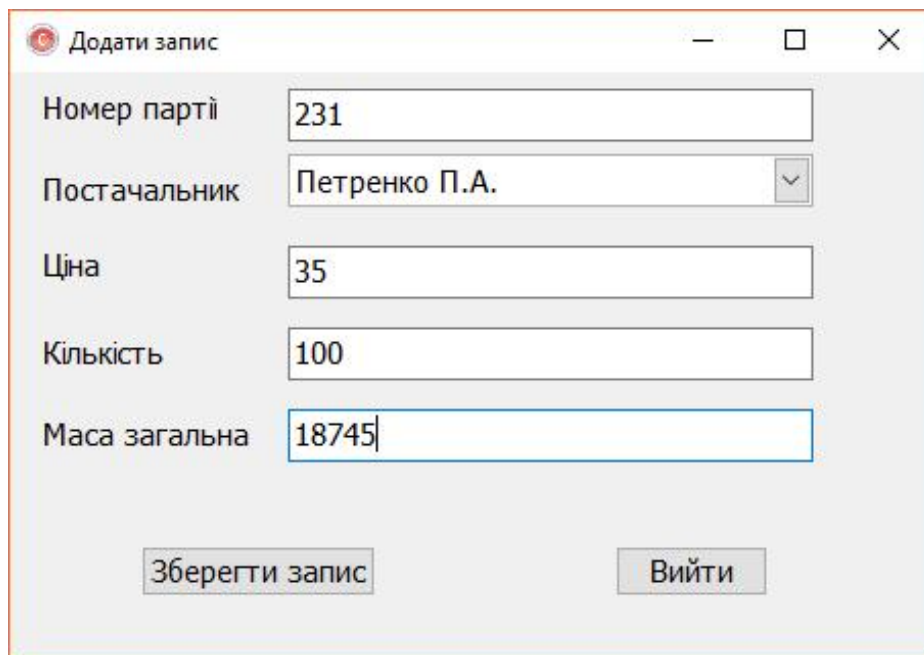


Рис. 2. Перше діалогове вікно C++ Builder проекту, призначене для додання запису в БД

В даній програмі передбачено ще одне діалогове вікно, яке необхідно використовувати після первинної обробки сировини. Тут передбачено заносення в БД інформацію про загальну масу півтуші свинини відповідної партії, як це показано на рис. 3.

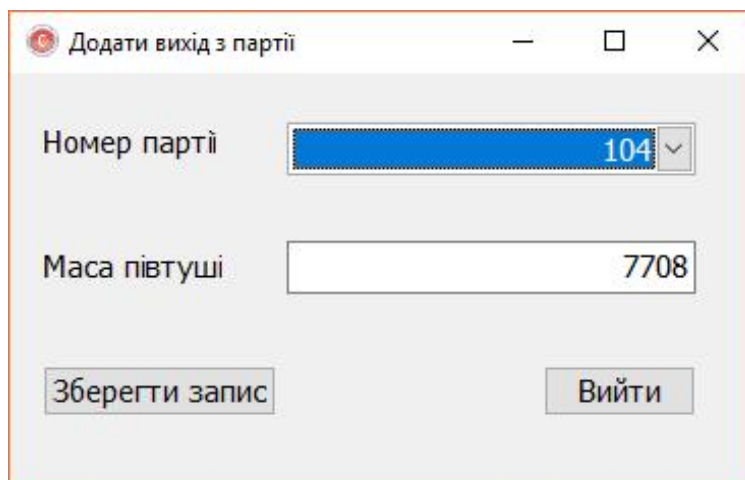


Рис. 3. Друге діалогове вікно C++ Builder проекту, призначене для додання запису в БД після первинної обробки сировини

Як видно з наведених рисунків інтерфейс програми є простим та зручним у користуванні. Розроблена програма підтримує такі операції:

- додання запису про партію в базу даних;
- додання, видалення та вивід інформації про постачальників;
- виведення статистики про отримання сировини за певний період;
- виведення рейтингу постачальників вигляді графіку рис.4;

Рейтинг постачальників формується за якісним показником виходу готової сировини, а саме ціною за один кілограм м'яса свинини. Тобто чим менший показник на графіку, що наведено на рис. 4 у вікні рейтингу постачальників, тим кращий постачальник продукції. Так, наприклад, станом на двадцять березня на першому місці є Петренко П.А. Також бачимо, що за даний період він пропонував нам середні, а то й найнижчі ціни порівняно з іншими постачальниками. Тому можна зробити висновок, що подальша співпраця з ним призведе до ефективного розвитку бізнесу.

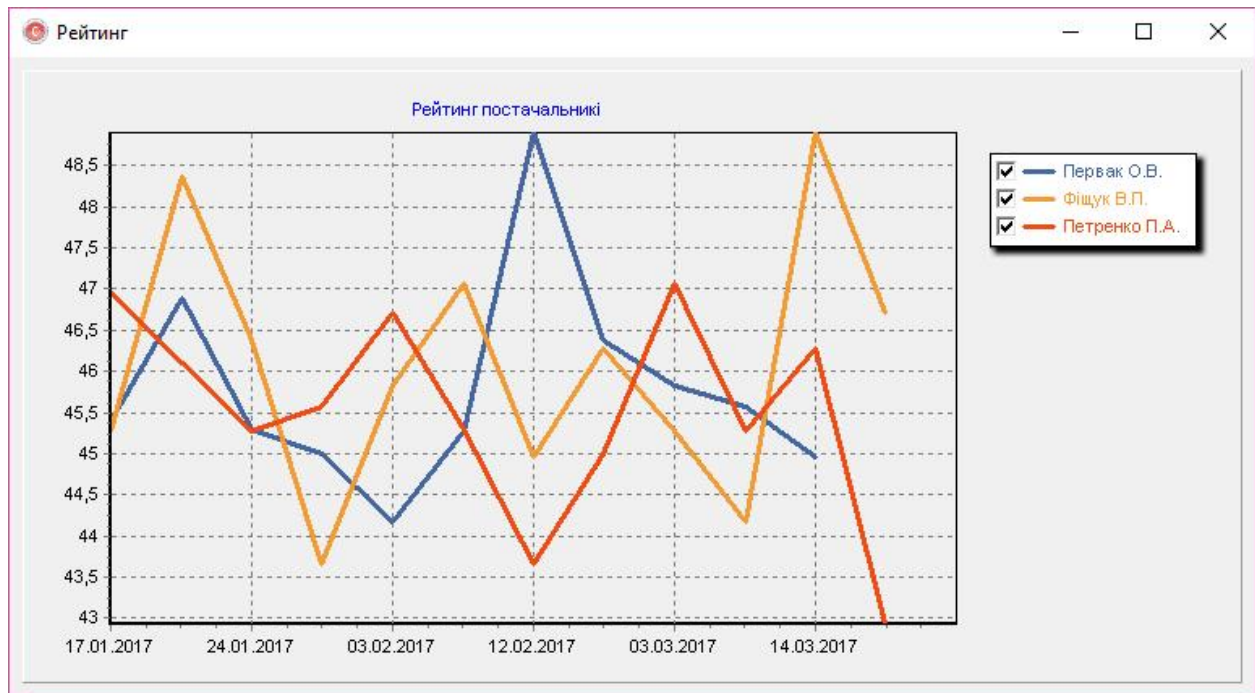


Рис. 4 Вікно рейтингу постачальників

Програма розроблена в середовищі C++ Builder XE, для роботи з базою даних використовувались такі компоненти ADO: ADOConnection, DataSource, ADOQuery, DBLookupComboBox, DBEdit, DBChart та інші.

База даних створена за допомогою СКБД MySQL. Архітектура бази складається з двох таблиць: перша – основна таблиця де зберігаються записи про кожну партію сировини, та друга – з інформацією про постачальників.

Оскільки створюючи програму невідомо скільки буде постачальників, адже ми можемо їх додавати чи видаляти. Графік рейтингу потрібно було робити динамічним, створюється він наступною функцією:

```
void __fastcall TRating_window::FormActivate(TObject *Sender) {
    for (int i=0; i<Rating_window->ADOQuery_tb_providers->RecordCount; i++) {
        TLineSeries *NewSeries = new TLineSeries(DBChart_rating);
        TADOQuery *NewQuery = new TADOQuery(Rating_window);
        NewQuery->Connection = ADOQuery_tb_providers->Connection;
        NewQuery->Close();
        NewQuery->SQL->Clear();
        NewQuery->SQL->Add("SELECT cost_pt, data_delivery FROM uistus.main_table WHERE
        provider="+Rating_window->ADOQuery_tb_providers->Fields->Fields[0]->AsString+");");
        NewQuery->Open();
        NewQuery->First();
        DBChart_rating->AddSeries(NewSeries);
        Rating_window->ADOQuery_tb_providers->FindNext();
        DBChart_rating->Series[i]->DataSource = NewQuery;
        DBChart_rating->Series[i]->Active = true;
        if (i==0)
            DBChart_rating->Series[i]->XLabelsSource="data_delivery";
        DBChart_rating->Series[i]->YValues->DateTime=false;
        DBChart_rating->Series[i]->YValues->Order=loNone;
        DBChart_rating->Series[i]->YValues->ValueSource="cost_pt";
        DBChart_rating->Series[i]->Title=Rating_window->ADOQuery_tb_providers->Fields->
        Fields[1]->AsString; }}
}
```

Висновки та перспективи подальшого дослідження. В результаті виконаної розробки програмного комплексу було реалізовано потребу малих підприємств у простій в користуванні та не дорогій програмі для визначення рейтингу постачальників сировини. Даний комплекс був спроектований в програмному середовищі C++ Builder XE та за СКБД MySQL. Тут взаємодія з базою даних здійснюється за допомогою технології ADO. Програма виводить рейтинг постачаль-

ників сировини, та загальну статистику про поставки залежно від дати. Для цього спроектовано два зручних діалогових вікна додавання записів з інформацією про партію сировини.

В майбутньому дану програму можна вдосконалити створивши додаткові критерії для рейтингу, чи легко змінити для підприємства з іншим видом діяльності. Також можна застосувати методи математичного аналізу, адже часто буває що показники в рейтингу різко змінюються, а таке вдосконалення дозволить більш точно визначити найкращого постачальника.

1. Архангельский А.Я. Программирование в С++ Builder / А.Я. Архангельский. – М. : ООО «Бином-Пресс», 2010 – 896 с.
2. Програма управління виробництвом [Електронний ресурс]. – http://usu.kz/app_proizvodstvo.php.
3. Fox Manager 2.5 ВРА – регламентация и анализ бизнес-процессов [Електронний ресурс]. – <http://www.fox-manager.com.ua/fox-manager-2-0-bpa.html>.
4. PriceMatrix [Електронний ресурс]. – <http://uk.softoware.net/business-office-software/download-pricematrix-for-windows.html>.

УДК 004.75: 629.735

Мусієнко М.П. д.т.н, проф., Журавська І.М., к.т.н, доцент, докторант
Чорноморський національний університет імені Петра Могили

СИСТЕМА КОНТРОЛЮ БЕЗПІЛОТНИХ АПАРАТІВ, ЗАСНОВАНА НА ВИКОРИСТАННІ МОБІЛЬНИХ ПРИСТРОЇВ

Мусієнко М.П., Журавська І.М. Система контролю безпілотних апаратів, заснована на використанні мобільних пристроїв. У статті запропонований метод контролю безпілотних літальних апаратів (БПЛА) за допомогою мобільних пристроїв, які здійснюють мобільний аналіз даних з БПЛА та оточуючого середовища. Метод базується на використанні сервісу Google Firebase як бази даних реального часу, засобу авторизації Клієнтів та Об'єктів системи контролю БПЛА, засобу синхронізації та мобільного аналізу даних. Розроблена відповідна програмно-апаратна система.

Ключові слова: контроль безпілотних апаратів, мобільна аналітика даних, Google Firebase.

Мусиенко М.П., Журавская И.Н. Система контроля беспилотных аппаратов, основанная на использовании мобильных устройств. В статье предложен метод контроля беспилотных летательных аппаратов с помощью мобильных устройств, осуществляющих мобильный анализ данных с БПЛА и окружающей среды. Метод основан на использовании сервиса Google Firebase как базы данных реального времени, средства авторизации клиентов и объектов системы контроля БПЛА, средства синхронизации и мобильного анализа данных. Разработана соответствующая программно-аппаратная система.

Ключевые слова: контроль беспилотных аппаратов, мобильная аналитика данных, Google Firebase.

Musiienko M.P., Zhuravska I.M. The control of unmanned vehicles, based on the use of mobile devices. The paper presents a method of drones' control via mobile devices that carry cell analysis of data from UAV and the environment. This method based on the use of Google Firebase services like real-time database, means of clients' and objects' authentication for UAV control system, means of synchronization and mobile data analysis. Defined development appropriate software and hardware system was developed.

Keywords: control of unmanned vehicles, mobile data analytics, Google Firebase.

Вступ. За останні роки тема використання безпілотних літальних апаратів (БПЛА) все частіше опиняється на слуху застосовно до різних галузей цивільно-військового призначення й в різнопланових напрямках людської діяльності, таких як управління врожайністю й диференційне обприскування рослин, доставка поштових відправлень, доставка продуктів, ведення моніторингових та оборонних дій в повітряному просторі [1–3].

Системи, які стосуються БПЛА, вимагають два види проектування. Перший з них - це проектування загальної моделі (прототипу) БПЛА або пристосування існуючих конструктивних рішень БПЛА до конкретних моніторингових або технологічних задач. Другий вид - проектування програмного забезпечення (ПЗ), яке забезпечуватиме взаємодію БПЛА (як об'єкту контролю) з пристроями, які узагальнюють отримані від БПЛА дані та координують дії БПЛА у разі збоїв автономного функціонування об'єкта або виникнення критичної ситуації.

Постановка наукової проблеми. Метою даної роботи є створення програмно-апаратної системи контролю БПЛА на основі мобільного аналізу даних з використанням кроссплатформного хмарного сервісу Google Firebase.

Аналіз останніх досліджень і публікацій.

На теперішній час накопичений немалий зарубіжний та вітчизняний досвід керування поодинокими БПЛА та зграями БПЛА з використанням інтегрованого підходу, за яким керування здійснюється з використанням наземних та повітряних трансферних вузлів [4, 5]. Сучасні системи керування БПЛА використовують для цього координати БПЛА, отримані від вбудованих в останні GPS-модулів. Але за 1 хв польоту похибка маршруту БПЛА, скоординованого за даними GPS-модуля, може становити до 50 м [6]. Проблемні питання точного досягнення кінцевої точки маршруту БПЛА під час використання сучасних супутникових технологій визначення координат повністю не вирішені [7].

Слід зауважити, що причинами виникнення неточних вимірів GPS-координат можуть бути:

1. Через штучне обмеження точності визначення координат цивільними GPS-навігаторами у межах 2–20 метрів [8, 9].

2. Хибне геометрія супутників. Виникає тоді, коли всі прийняті супутники згруповані близько один до одного, чи вибудовані в лінію щодо положення приймача, що у свою чергу не забезпечує необхідну розбивку сигналу на так званий трикутник, від чого точність обчислення

координат стає менш надійною. Таке непередбачуване зміщення положення супутників на орбіті відбувається через різні збурюючі фактори та приводять до похибок 0,6–10 м [10].

3. Сигнали перед тим, як досягти GPS-приймача, можуть бути відбиті від високих будинків чи інших перешкод, тим самим або збільшується відстань сигналу, або сигнал попросту збивається, зменшуючи точність.

Зважаючи на те, що сучасні міні-БПЛА живляться від акумуляторів на борту, ємність яких забезпечує тривалість польоту від 7 до 28 хв [11]. Отже, можна зробити висновок, що для забезпечення якості виконання завдання необхідно здійснювати додаткове керування польотом БПЛА з коригуванням маршруту останнього до точки цілі. Поширеним підходом до вирішення питань такого керування є використання бездротових сигналів від мобільних пристроїв.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Пропонується спосіб контролю БПЛА за допомогою мобільних пристроїв, які здійснюють мобільний аналіз даних з БПЛА та оточуючого середовища, що базується на використанні сервісу Google Firebase як БДРЧ (бази даних реального часу), засобу авторизації Клієнтів та Об'єктів системи контролю БПЛА, засобу синхронізації даних [12]. Зазначений безкоштовний хмарний сервіс є універсальною платформою для побудування Android- та iOS- мобільних додатків. Перевагою саме цього сервісу є стабільна робота навіть при збоях інтернет-зв'язку. Слід зауважити, що при використанні Клієнтів з різних платформ необхідно зареєструвати два облікових записи в Google (окремо для Android та для iOS).

На рис. 1 наведені можливі взаємозв'язки між Клієнтами та Об'єктами розробленої системи з використанням мобільних пристроїв для контролю БПЛА.

У Firebase SDK сервер має вбудований метод для створення користувацьких ідентифікаторів. Firebase створює відповідний унікальний ідентифікатор (рис. 2), який ідентифікує пристрій у мережі і надає певний доступ до ресурсів, а саме до бази даних у реальному часі Firebase Storage. Тобто, маємо ідентифікатор Об'єкта, за допомогою якого він ініціалізується Клієнтом.

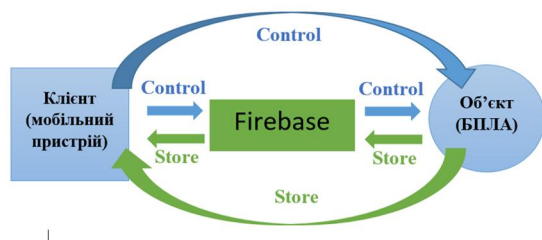


Рис. 1. Взаємодія мобільного пристрою з БПЛА через сервіс Google Firebase (авторська розробка)

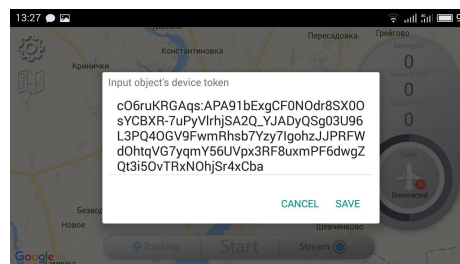


Рис. 2. Ідентифікатор Об'єкта в налаштуваннях Клієнта (авторська розробка)

На доступній хмарі Google Firebase реалізована структура подієвого збору даних в розрізі користувачів (Клієнтів та Об'єктів), які пройшли авторизацію в названій хмарі.

Сегменти користувачів (так звані «аудиторії») розподіляються як за попередньо налагодженими параметрами (зона знаходження, технічні дані платформ БПЛА/мобільний пристрій тощо), так і за подієвими даними.

Наприклад, на рис. 3а наведена «аудиторія», створена на Google Firebase, яка складається з мобільного пристрою (Клієнт, зображений червоною крапкою) та двох керованих БПЛА (об'єктів, переміщення яких зображене зеленими крапками). Зазначена «аудиторія» сформована за географічним принципом знаходження в одному квадраті місцевості.

На рис. 3б наведена «аудиторія», сформована за ознаками платформ (Android).

Таким чином, унікальний рядок символів може бути переданим іншим членам «аудиторії» або через SMS-сервіс, або іншими соціальними мережами (коли відсутня можливість користування сервером). Після цього клієнт встановлює з'єднання з об'єктом и отримує доступ до ресурсів Firebase, де надається можливість зберігати дані об'єкта у реальному часі.

Після авторизації в Google Firebase, Об'єкт (БПЛА) кожні 5 секунд відправляє свої дані місцезнаходження Клієнтові свої «аудиторії». При цьому не має значення наявності або відсутності безпосереднього зв'язку між Клієнтом та Об'єктом (Об'єктами), кого контролює цей

Клієнт – засоби мобільного аналізу даних Google Firebase забезпечать передачу коригуючих команд саме необхідному Об'єкту.

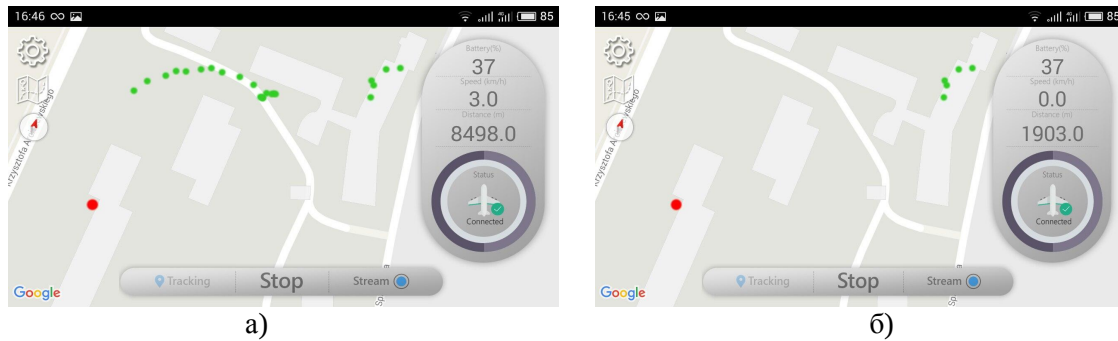


Рис. 3. «Аудиторія» Google Firebase, сформована за географічним принципом (а)
та за принципом платформи (б)
(авторська розробка)

З правого боку екрану Клієнта розташований індикатор поточного заряду акумулятора Об'єкта (у відсотках), швидкість Об'єкта (км/год) та дистанція між Клієнтом та Об'єктом (в метрах).

У запропонованій системі контролю відтворений розроблений протокол взаємодії між Клієнтом та кожним з Об'єктів з врахуванням показників цілого набору пристроїв на борту БПЛА: акселерометра, барометра, гіроскопів, магнітографа, GPS-модуля, GSM-модуля тощо.

За результатами мобільного аналізу сукупності наведених показників Клієнтом може бути змінені, наприклад, координати фінішу Об'єкта – широта і довгота (latitude and longitude). Діаграма класів для програмної реалізації зазначеного завдання наведена на рис. 4.

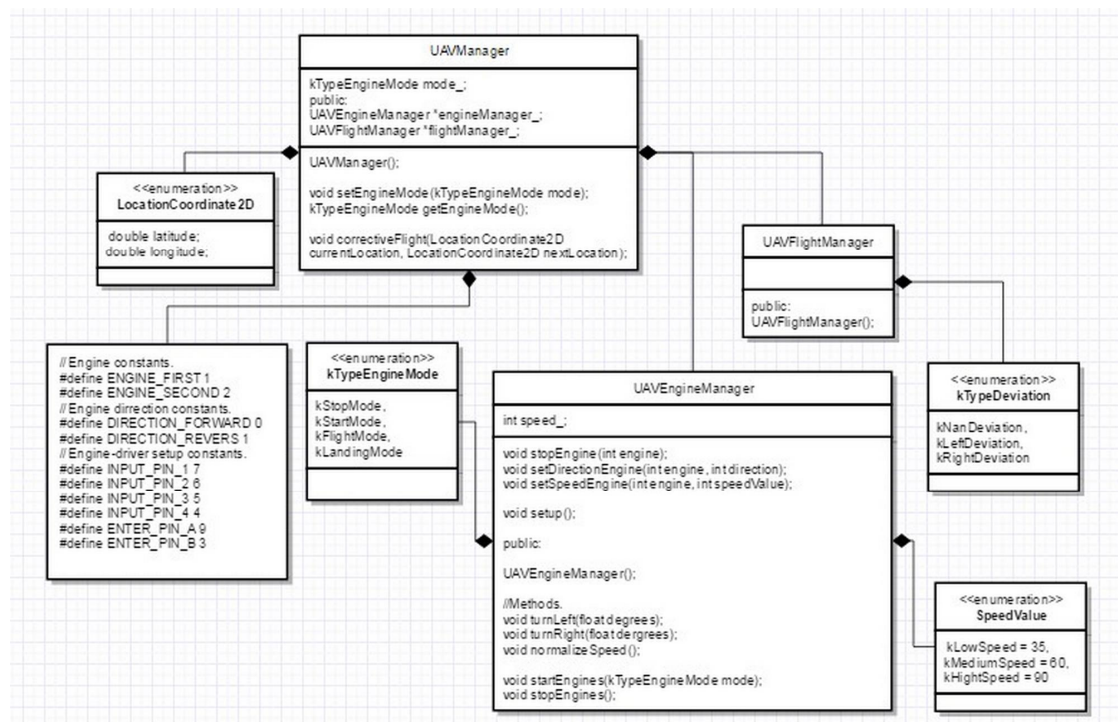


Рис. 4. UML-діаграма класів зміни координат фінішу при керуванні БПЛА
(авторська розробка)

Приклад оформлення протоколу взаємодії GPS-модуля Об'єкту з Клієнтом наведений на рис. 5, деталізація компонент протоколу представлена у табл. 1.

>\$GPRMC,170840.00,A,5445.33016,N,1704.38950,E,1.6198,,310317,,,A*7D

Рис. 5. Нотація протоколу взаємодії Клієнта з Об'єктом при зміні координат фінішу БПЛА
(авторська розробка)

Таблиця 1. Деталізація компонент зміни GPS-координат фінішу при керуванні БПЛА
(авторська розробка)

170840.00	Час UTC: 17 год 08 хв 40,00 с
A	Статус достовірності: A – достовірні дані, V – недостовірні дані
5509.68339	Широта: 54 град. 45,33016 хв
N	N – північ, S – південь
06125.49498	Довгота: 17 град. 04,38950 хв
E	E – схід, W – захід
1.6198 (відповідає 3 км/год на рис. 2а)	Горизонтальна швидкість (вузлів/год)
–	Напрямок курсу відносно півночі (градуси)
310317	Дата: 31 березня 2017 р.
–	Магнітне відхилення (градуси)
–	Напрямок відхилення: E – східний, W – західний
A	Режим: A – автономний, D – диференціальний, E – апроксимація, N – недостовірні дані
*7D	Контрольна сума

Коли Об'єкт прибув до цілі, на екрані Клієнта з'являється відповідне повідомлення (рис. 6).

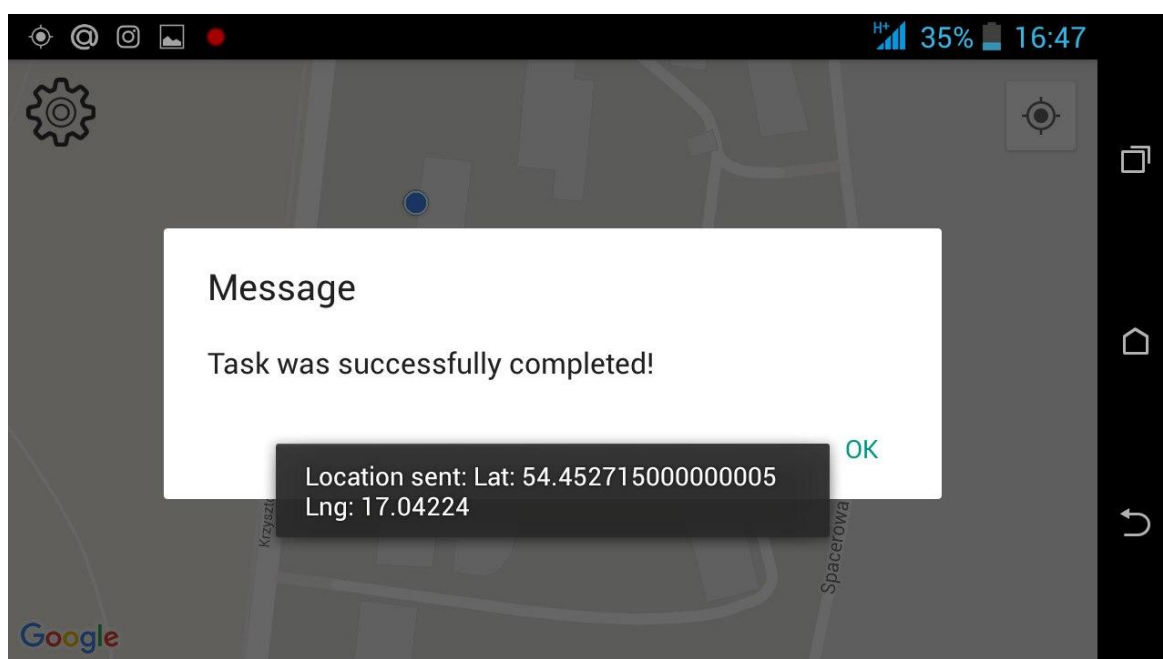


Рис. 6. Повідомлення на екрані Клієнта після виконання завдання керованим БПЛА
(авторська розробка)

Із співставлення координат широти та довготи рис. 5, табл. 1 та рис. 6 можна побачити, що наявне відхилення між заданими системою керування координатами та координатами фінішу після виконання завдання. Таке відхилення є наслідком неідеальної роботи GPS-модуля БПЛА, в результаті чого відстань рахується з похибками.

Висновки

Проаналізовані та узагальнені причини похибки маршруту БПЛА у разі прокладання маршруту на підставі даних з GSM-модулю БПЛА. Доведена необхідність коригування маршруту БПЛА.

Запропоновано спосіб контролю БПЛА з використанням мобільних пристроїв та застосуванням хмарного сервісу Google Firebase, який надає інтегровані послуги авторизації мобільних пристроїв та БПЛА, синхронізації та аналізу даних, що зберігаються у БДРЧ зазначеного сервісу. Розроблена відповідна програмно-апаратна система. Використання властивостей Google Firebase у розробленій програмно-апаратній системі дозволяє позбавитись від зберігання отриманої моніторингової інформації як на самому БПЛА, так і на пристрої контролю БПЛА, залишаючи натомість інформацію про залежність цих даних у зазначених пристроях.

Такий підхід суттєво підвищує безпеку поточної інформації щодо функціонування БПЛА у разі перехоплення Об'єкта сторонніми особами.

Крім того, надійна платформа та технології Google забезпечують доступність, інтеграцію в інші Google-сервіси (Google Maps ті ін.), а також масштабованість Об'єктів у зграї БПЛА.

Напрямом подальшого дослідження є розробка способів коригування похибок визначення відстані від мобільного пристрою до керованого БПЛА, спричиненої неточністю результатів функціонування GPS-модулів БПЛА.

1. Mesas-Carrascosa F.-J. Assessing optimal flight parameters for generating accurate multispectral orthomosaics by UAV to support site-specific crop management / F.-J. Mesas-Carrascosa, J. Torres-Sánchez, I. Clavero-Rumbao and al. // Remote Sensing. – 2015. – No. 7. – Pp. 12793-12814. DOI: 10.3390/rs71012793
2. Colomina I. Unmanned aerial systems for photo-grammetry and remote sensing: A review / I. Colomina, P. Molina // ISPRS Journal of Photogrammetry and Remote Sensing. – 2014. – Vol. 92 – Pp. 79–97. DOI: 10.1016/j.isprsjprs.2014.02.013.
3. Митрахович М. М. Беспилотные летательные аппараты: Методика сравнительной оценки боевых возможностей / М. М. Митрахович, В. И. Силков, А. В. Самков, Х. В. Бурштынская и др. ; под общ. ред. В. И. Силкова. – Киев : ЦНИИ ВВТ ВС Украины, 2012. – 288 с.
4. Teh S. Experiments in Integrating Autonomous Uninhabited Aerial Vehicles (UAVs) and Wireless Sensor Networks [Електронний ресурс] / S. Teh, L. Mejias, P. Corke, W. Hu // Proceedings of the Australasian Conference on Robotics and Automation, Canberra, Australia, 3–5 December 2008. – Режим доступу : <http://www.araa.asn.au/acra/acra2008/papers/pap118s1.pdf> (дата звернення 14.04.2017).
5. Musiyenko M. P. The Principles of the Cyber-Physical Components' Organization Based on the Methods of the Multi-Agent Interaction of the Moving Objects / M. P. Musiyenko, I. M. Zhuravska, I. S. Burlachenko, O. O. Denysov // Advances in Cyber-Physical Systems. – 2016. – Vol. 1, No. 1. – Pp. 48–57.
6. Шульц Р. В. Дослідження систематичних похибок інерціальної навігаційної системи при аерофотозніманні з безпілотних літальних апаратів / Р. В. Шульц, П. Д. Крельштейн, І. А. Маліна // Scientific Journal «ScienceRise». – 2015. – №9/2(14). – С. 6–18. DOI: 10.15587/2313-8416.2015.49183
7. Савчук С. Г. Проблемні питання під час використання сучасних супутникових технологій визначення координат // Геодезія, картографія і аерофотознімання / Нац. ун-т «Львівська політехніка». – 2007. – Вип. 69. – С. 20–33.
8. Найман В. С. Всё о GPS-навигаторах / В. С. Найман, А. Е. Самойлов, Н. Р. Ильин, А. И. Шейнис. – М. : НТ Пресс, 2005. – 392 с.
9. Збільшення точності визначення GPS координат [Електронний ресурс] // Веб-сайт Intelli: система контролю та моніторингу транспорту в Україні. – Режим доступу : <http://intelli.com.ua/ua/statti/zbilshennia-tochnosti-vyznachennia-gps-koordynat.html> (дата звернення 14.04.2017).
10. Петрівний О. І. Дослідження точності супутникових навігаційних систем // Наукові читання-2013 : наук.-теорет. зб. / Житомир. нац. агроєкологіч. ун-т. – Житомир : ЖНАЕУ, 2013. – Том 1. – С. 367–373.
11. Austin R. Unmanned aircraft systems: UAVS design, development and deployment. – Chichester, West Sussex, UK : Wiley, 2010. – 365 p.
12. Синиця І. Firebase Analytics: настраиваем аналитику мобильных приложений [Електронний ресурс] // Netpeak Blog: веб-аналитика. – Режим доступа : <https://netpeak.net/ru/blog/nastraivaem-analitiku-mobilnyh-prilozheniy-s-pomoshchyu-firebase-analytics/> (дата обращения 14.04.2017).

УДК 004.77

Орлова М.М. к.т.н, доцент, Мосійчук І.В. магістрант

Національний технічний університет України «Київський політехнічний інститут ім.
І.Сікорського»

МОДИФІКОВАНИЙ АЛГОРИТМ ПОШУКУ НАЙКОРОТШИХ МАРШРУТІВ ПЕРЕДАЧІ ІНФОРМАЦІЇ В ОДОРАНГОВИХ МЕРЕЖАХ

Орлова М.М., Мосійчук І.В. Модифікований алгоритм пошуку найкоротших маршрутів передачі інформації в одорангових мережах. У статті пропонується модифікований алгоритм пошуку найкоротших маршрутів з метою пришвидшення передачі інформації в одорангових мережах та його порівняння з базовим алгоритмом. Даний алгоритм використовує метод «Rejection Sampling» для визначення проміжкових вузлів шляху передачі даних.

Ключові слова: одорангова мережа, децентралізований алгоритм, граф «Світ тісний».

Mariia Orlova, Ihor Mosiichuk modified method of searching the shortest rout of transfer data for peer-to-peer network. This paper describes a algorithm of routing for peer-to-peer corporate network based on decentralized algorithm. Finding the shortest rout of transfer to increase speed of exchanging resources in this network and comparing it with basic algorithm. This algorithm use «Rejection Sampling» method for finding interim peers of transfer's rout.

Keywords: peer-to-peer network, decentralized algorithm, Small-world web.

Вступ. Стрімкий розвиток прогресу надає безліч можливостей для комунікації сучасної людини. Мережа Інтернет охоплює весь світ і таким чином пришвидшує взаємодію між людьми на великих відстанях. Виникають нові методи передачі даних, основною вимогою до яких є забезпечення високої швидкодії. Виникають нові види мереж, протоколи та алгоритми взаємодії між відправником та отримувачем інформації. При цьому чим краще структуровано мережу, тим швидша взаємодія.

Одорангові мережі набувають значного поширення. Головною особливістю таких мереж – є робота без центрального сервера, тобто обмін даними між користувачами відбувається на пряму між ними, по рівнозначним зв'язкам. В таких мережах можна спостерігати феномен «Світ тісний». З алгоритмічної точки зору це мережі, в яких передача повідомлень між модулями відбувається з використанням тільки локальної інформації, тобто інформації про вузли, які з'єднані безпосередньо з сусідніми вузлами (модулями) мережі.

В даній роботі розглядаються особливості локальної мережі, побудованої на основі мережі Джона Клайнберга [2], та її функціонування з використанням децентралізованого алгоритму.

Постановка задачі. Метою роботи є розробка модифікованого способу передачі пакетів даних між вузлами корпоративної мережі та дослідження його характеристик.

Теоретичні відомості

Феномен «Світ тісний» виник в соціальній науці, він полягає в тому, що всі люди пов'язані між собою коротким ланцюгом знайомств. Згодом це явище почали вивчати і в інших науках, теорії графів, теорії мереж та використовувати в техніці та ІТ технології.

В теорії графів граф «Світ тісний» має такі властивості: дві довільні вершини a та b , які не є суміжними, зв'язані між собою за допомогою невеликої кількості переходів через інші вершини. На рис.1 представлено один з прикладів такого графу.

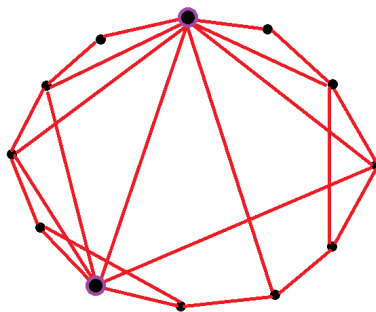


Рис. 1. Граф «Світ тісний»

Характеристика мережі, що описується за допомогою таких графів, полягає в наступних критеріях:

- структура мережі повинна бути відома для сусідніх вузлів;
- мережа містить короткі шляхи до більшості віддалених вузлів;
- зв'язки між вузлами є рівнозначними, тобто мають однакову вагу.

Отже, для побудови шляхів до віддалених вузлів, які не мають безпосередніх зв'язків з даним вузлом (модулем) може бути використана маршрутна інформація від сусідніх вузлів.

Структуру такої мережі можна представити двовимірним графом, вершини цього графа характеризують вузли, а ребра – канали з'єднання, при цьому ребра є ортогональними.

Розглянемо двовимірну мережу, граф якої представляє квадратну матрицю розміром $n \times n$, де $n=6$ (рис. 2).

Розглянемо двовимірну мережу, граф якої представляє квадратну матрицю розміром $n \times n$, де $n=6$ (рис. 2). В цій мережі кожен вузол з'єднаний з r локальними сусідніми вузлами ($r > 1$) і не з'єднаний з жодним віддаленим вузлом ($q = 0$). А всі зв'язки між вузлами мають однакову вагу.

В такому випадку манхетенська метрика $d((i, j), (k, l))$ між двома вузлами з координатами (i, j) і (k, l) , тобто відстань між двома вузлами мережі, що дорівнює сумі різниць координат вузлів, взятих по модулю, буде визначатися за наступною формулою

$$d((i, j), (k, l)) = |k - i| + |l - j|.$$

Фактично, ця відстань дорівнює кількості кроків між вузлами при передачі обраним маршрутом.

У випадку коли $q > 0$, наприклад, $q=2$ (рис.3), для вузла u вузли v та w є віддаленими модулями (контактами). Тобто вузол u має локальну інформацію про 4 сусідні вузли та 2 віддалені модулі.

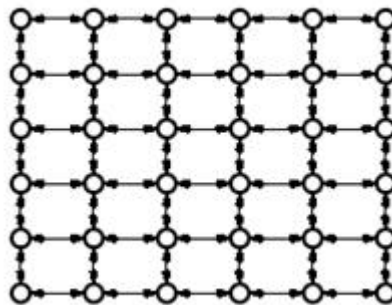


Рис. 2. Двовимірні сітка з $n = 6$, $q = 0$

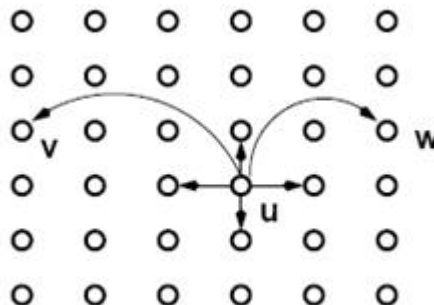


Рис. 3. Контакти вузла u при $q = 2$, де v та w - віддалені контакти.

Географічна інтерпретація такого підходу досить проста: вузли «знаходяться в мережі» і знають в певній кількості кроків у всіх напрямках своїх сусідів (сусідні вузли), також у них є кілька «знайомих», які знаходяться далі. Вважаючи p і q фіксованими константами, розглядають одне параметричне сімейство мереж. Тоді отримуємо рівномірний розподіл віддалених контактів (вони вибираються незалежно від своєї позиції в мережі). При збільшенні кількості віддалених модулів, контакти вузла стають все ближче один до одного. Таким чином, параметр q є базовим і вимірює, наскільки логічно розгалуженою є дана мережа. При $q=0$ маємо рівномірний розподіл віддалених вузлів.

Алгоритмічна компонента передачі даних в такій мережі ґрунтується на експерименті Мілграма [3], який полягає в наступному: для двох випадкових вузлів мережі виконується пошук найкоротших шляхів передачі інформації (з найменшою кількістю кроків), для чого застосовуються децентралізовані алгоритми.

Децентралізовані алгоритми – це механізми, за допомогою яких повідомлення передається послідовно від одного вузла (поточного) до іншого використовуючи локальну інформацію. На кожному кроці поточний вузол на маршруті передачі має наступні відомості:

- множина локальних контактів;
- місце розташування пункту призначення (кінцевого вузла, отримувача) в мережі;
- місце розташування вузлів, що вже використовувались для передачі інформації.

Час доставки для децентралізованого алгоритму - кількість кроків, яку виконує алгоритм для доставки повідомлення від відправника до отримувача. Так як для пошуку маршруту передачі між віддаленими вузлами використовується тільки локальна інформація то це є суттєвим обмеженням, тобто недоліком даного алгоритму.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Модифікований алгоритм пошуку найкоротших шляхів маршруту. Даний алгоритм передбачає для пошуку необхідного шляху передачі використання не тільки локальної інформації, а й відомості про деякі віддалені вузли, що дозволить скоротити час як пошуку маршруту, так і передачі даних між віддаленими вузлами.

Для цього для кожного вузла додаються спеціальні ярлики, які вибираються випадково.

Манхетенська відстань $d(a,b)$ між $a=(a_x;a_y)$ та $b=(b_x;b_y)$ буде дорівнювати $d(a,b) = |a_x - b_x| + |a_y - b_y|$, а ймовірність вибору вузла b в якості вузла подальшої передачі $1/d(a,b)^q$ для деякого $q>0$ яке не залежить від a та b .

Після отримання даної інформації за допомогою методу «Rejection sampling» визначаємо наступний вузол на шляху передачі. Він вибирається по критерію найменшої ймовірності, це дозволяє зменшити шлях передачі і отримати менший час передачі.

Модифікація децентралізованої маршрутизації полягає в тому що вибирається конкретний вузол, що є найближчим за манхетенською відстанню до пункту призначення. При цьому шлях передачі пакетів даних між двома випадково вибраними вузлами мережі найкоротшим.

Результати роботи запропонованого модифікованого алгоритму для двомірної мережі 4x4 при випадковому виборі вузлів відправлення та отримання інформації та її порівняння з базовим з базовим алгоритмом представлено на рис. 4.

Для визначення характеристик розробленого алгоритму було проведено 100 експериментів для випадково обраних вузлів відправника та отримувача інформації. При проведенні дослідження визначалась залежність кількості знайдених найкоротших маршрутів від кількості проміжних вузлів.

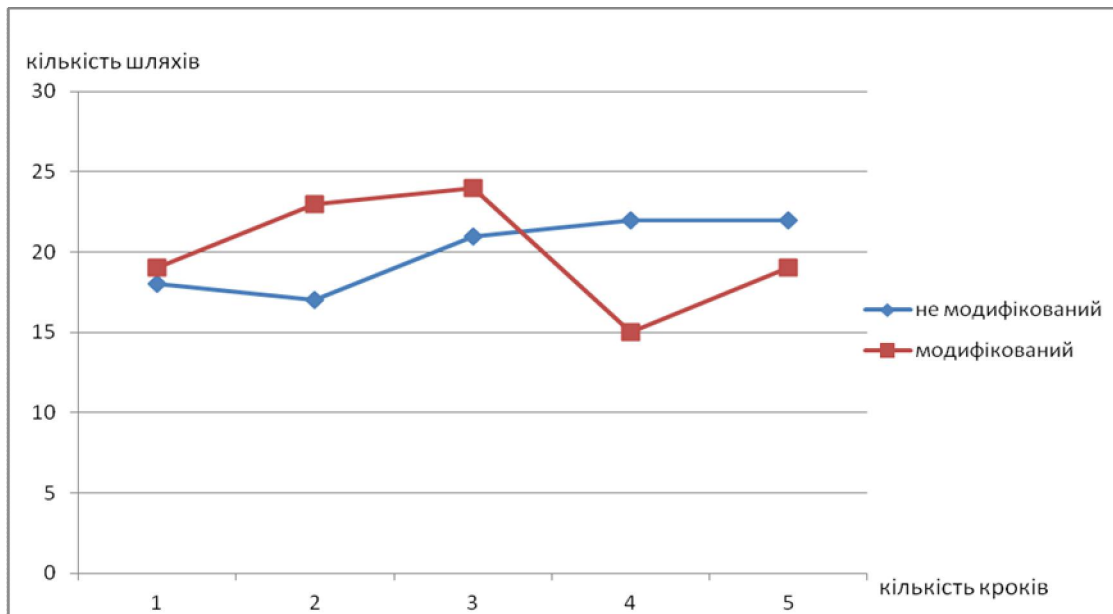


Рис. 4. Графік Порівняння модифікованого методу та базового (не модифікованого)

Висновки

В роботі проаналізовано децентралізований алгоритм передачі даних у одноранговій мережі, з рівнозначними зв'язками між вузлами, побудований на основі графу «Світ тісний». Запропонована та досліджена модифікована версія даного алгоритму з використанням методу «Rejection sampling».

Проведені дослідження показують, що при виборі 100 випадкових пар взаємодіючих вузлів кількість найкоротших шляхів довжиною 1 крок (1 проміжний вузол) дорівнює 19 (18 – для базового); довжиною 2 кроки відповідно 23 та 17; 3 кроки – 21 та 24; 4 кроки – 22 та 15; 5 кроків (вузлів) – 22 та 19.

Тобто кількість найкоротших шляхів менша у модифікованого алгоритму. А це значить що він є більш швидким способом передачі даних в такій мережі.

1. J. Kleinberg. Small-world Phenomena and the dynamics of information. Advances in Neural Information Processing Systems (NIPS). - 2001 p. 14.
2. J. Kleinberg. Query Incentive Networks. Proc 46 the IEEE Symposium of Foundation of Computer Science.- 2005.
3. S.Milgram. The small world problem. Psychology today1.- 1967.
4. J. Kleinberg. Small-world Phenomenon and decentralized search. Siam news. - 2004. Volume 37, number 3.
5. Rejection sampling [Електроний ресурс] Режим доступу: https://en.wikipedia.org/wiki/Rejection_sampling

УДК 004.52

Поліщук М.М., Данилюк Л.М., Буденко Ю.В.
Луцький національний технічний університет

ЕЛЕКТРОННИЙ ТЕРМОМЕТР ІЗ ГОЛОСОВИМ ПОВІДОМЛЕННЯМ ТЕМПЕРАТУРИ

М.М. Поліщук, Л.М. Данилюк, Ю.В. Буденко. Електронний термометр із голосовим повідомленням температури. У статті представлено огляд електронних пристроїв сучасного будинку, класифікацію вимірювальних приладів, продемонстровано: друковану плату пристрою, принципову схему та структурну схему мікроконтролера, розроблено вимірювальний прилад температури із голосовим повідомленням.

Ключові слова: термометр, AVR, AT7294C, мікроконтролер, друкована плата, компатор, вимірювальний прилад, голосове управління

Н.Н. Полищук, Л.Н. Данилюк, Ю.В. Буденко. Электронный термометр с голосовым сообщением температуры. В статье представлен обзор электронных устройств современного дома, классификацию измерительных приборов, продемонстрировано печатную плату устройства, принципиальную схему и структурную схему микроконтроллера, разработаны измерительный прибор температуры с голосовым сообщением.

Ключевые слова: термометр, AVR, AT7294C, микроконтроллер, печатная плата, компатор, измерительный прибор, голосовое управление

M.M. Polishchuk, L.M. Danulyk, J.V. Bydenko. Electronic thermometer temperature voice message. An overview of modern electronic devices at home, classification instrumentation, shown to the printed circuit board unit schematic diagram and block diagram of a microcontroller designed gauge the temperature of the voice message.

Keywords: thermometer, AVR, AT7294C, microcontroller, PCB, kompator, gauge, voice control

Вступ. Сучасний будинок важко уявити без електричних пристроїв – від вже звичайної “економічної” освітлювальної лампи до складних комп'ютерних систем. Для спрощення керування ними та об'єднання в одну централізовану мережу розроблені різноманітні системи. Найпростіші електроприлади, на кшталт ламп освітлення, обігрівачів та електричні розетки, керуються простим вмиканням-вимиканням за допомогою реле [1]. Складніші пристрої, які містять в собі мікроконтролер чи мікропроцесор, отримують керуючі сигнали по одному із протоколів обміну та виконують вказану дію [2]. Вимірювальний прилад – засіб вимірювань, в якому створюється візуальний або акустичний сигнал вимірювальної інформації. Основне їх призначення — візуальний показ вимірюваної інформації.

В наш час вимірювальні прилади класифікуються за наступними ознаками :

- за видом подання інформації;
- за видом структурної схеми;
- за типом обчислювального пристрою;
- за характером використання;
- за видом енергії що використовується;

У зв'язку з широким розвитком комплексної автоматизації виробничих процесів і експериментальних досліджень із застосуванням ЕОМ зростають відповідні вимоги до вимірювальної техніки:

- підвищення точності, швидкодії, чутливості при вимірюванні величин, які змінюються;
- здійснення повної автоматизації складних процедур прямих, непрямих, сукупних і сумісних вимірювань;
- видача результатів вимірювань у кодованій формі безпосередньо інформаційно-вимірювальній системі;

Ці завдання вирішують цифрові вимірювальні прилади, поширення яких зумовлене наступними перевагами цих засобів вимірювання порівняно з аналоговими вимірювальними приладами. Швидкодія до сотень, мільйонів вимірювань у секунду. Висока інформаційна спроможність цифрових відлікових приладів значно підвищує здатність оператора до сприйняття інформації. Тому ЦВП мають крім цифрових відлікових пристроїв вихід у вигляді коду для зв'язку з ЕОМ, пристроями пам'яті або принтерами. Висока точність, яка перевищує при наявності автоматичної обробки результатів вимірювань точність цифрових приладів ручного зрівноважування. Відсутність суб'єктивної похибки результату вимірювання, наявність якої (при обмеженій довжині шкали) лімітує максимально досягнутою точністю аналогових показуючих приладів. Наявність виходу у вигляді кодового сигналу, зручного для цифрової обробки, а також

для запам'ятовування і передачі. Можливість автоматичної калібровки і автоматичного введення поправки для зменшення систематичної похибки. Можливість усереднення результатів вимірювання для зменшення випадкових похибок.

Постановка проблеми. Цифровими називаються такі вимірювальні прилади, в яких вимірювана величина автоматично в результаті квантування, дискретизації, порівняння, цифрового кодування і відповідних обчислень постає у вигляді коду, що виражає значення цієї величини.

Термометри із голосовим сповіщенням є актуальними як і в побуті так і на підприємствах чи установах де потрібен постійний контроль за температурою, також він в стані в нагоді для людей із обмеженими можливостями. Головною його перевагою таких пристроїв є не тільки їх універсальність, а також їх безпечність у використанні, оскільки у пристрої не використовується ртуть, а у корпусі немає скла яке можна розбити. Більшість термометрів із голосовим сповіщенням керується за допомогою кнопки на головному блоці приладу, оскільки це найпростіше та найдешевше рішення. Головний блок та датчик температури з'єднані за допомогою кабелю або радіо передавача. Розробка бюджетного пристрою вимірювання температури із голосовим повідомленням із можливістю модернізації, може стати новим рішенням у розв'язанні проблеми.

Аналіз існуючих способів. На сьогоднішній день промисловістю виробляється велика кількість різноманітних вимірювальних приладів, проте переважна більшість має велику вартість.

На українському ринку цифрових вимірювальних приладів із голосовою або візуальною подачею температури основну ланку посів соціально-комерційний проект Харківський соціально-реабілітаційний центр незрячий і підприємство об'єднання громадян «Лемур», який зображено на рис. 1. Вони виготовляють різноманітні пристрої для полегшення побуту, спостереження та допомоги в повсякденному житті. Це пристрої такі як: термометри, годинники, телефони, ваги, калькулятори та ін.



Рис. 1. Балакучий термометр "Градус — 05"

Проаналізувавши доступний ринок для покупця, легко зробити висновок, що доступні термометри мають високу вартість. Розроблений пристрій на базі мікроконтролера AVR AT7294C, має високі переваги перед конкурентами. Дана розробка дозволяє суттєво зменшити вартість приладу.

Виклад основного матеріалу. Пропонується вимірювальний прилад, який складається з таких двох блоків: головний блок та блок датчика температури, який представлено на рис. 2. В якості керувального пристрою було обрано мікроконтролер AVR AT7294C – представник сімейства однокристальних мікроконтролерів американської фірми Atmel. Структурна схема мікроконтролера представлена на рис. 3



Рис. 2. Загальний вигляд електронного термометра з голосовим управлінням

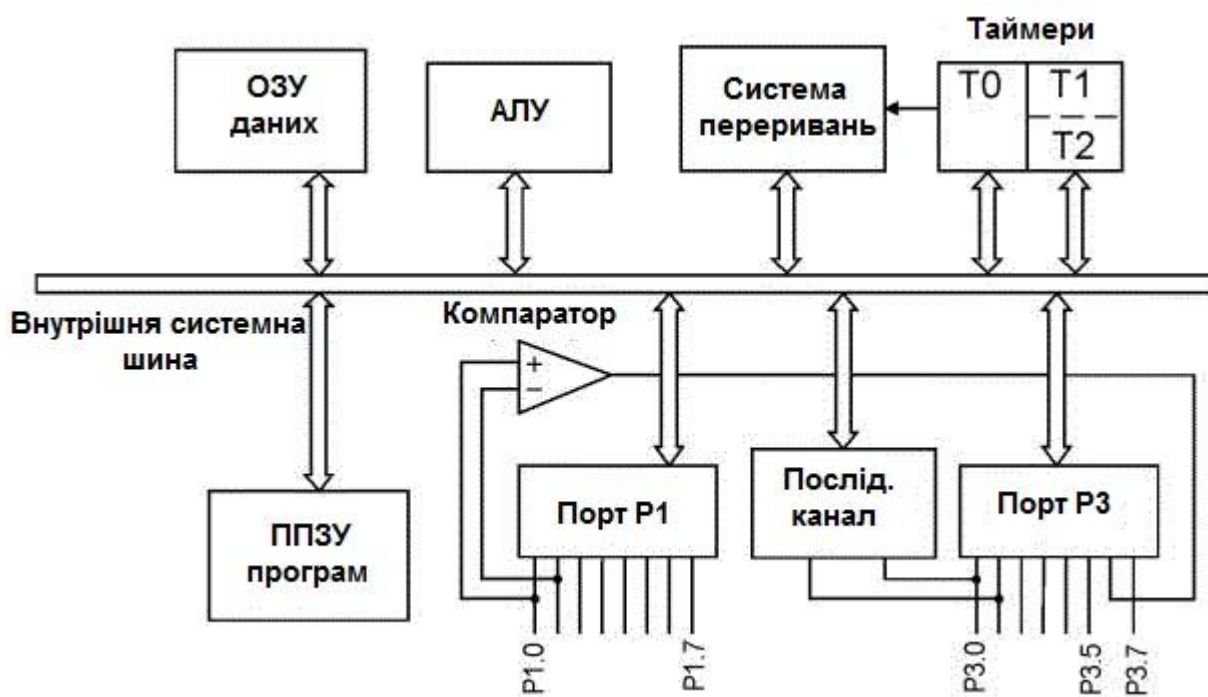


Рис. 3. Структурна схема мікроконтролера AT7294C

Мікросхема виконана в стандартному DIP - корпусі і має 20 виводів. Основна перевага даного мікроконтролера - це сумісність за системою команд з широко поширеною мікросхемою фірми Intel - MCS-51 (радянський аналог 1816BE51). Розробники ставили завдання створити мікросхему, максимально сумісну зі своїм аналогом, але при цьому має менші габарити і більш зручну в застосуванні.

У контролері відсутні режими роботи що потребують зовнішніх зв'язків із мікросхемами, а також вбудований тактовий генератор, в якості пам'яті команд застосоване електрично перепрограмований постійний запам'ятовуючий пристрій. Також до переваг мікросхеми

AT89C2051 відноситься наявність режиму захисту програми, в пам'ять програм мікросхеми від несанкційованого копіювання. Принципова схема пристрою зображена на рис. 4.

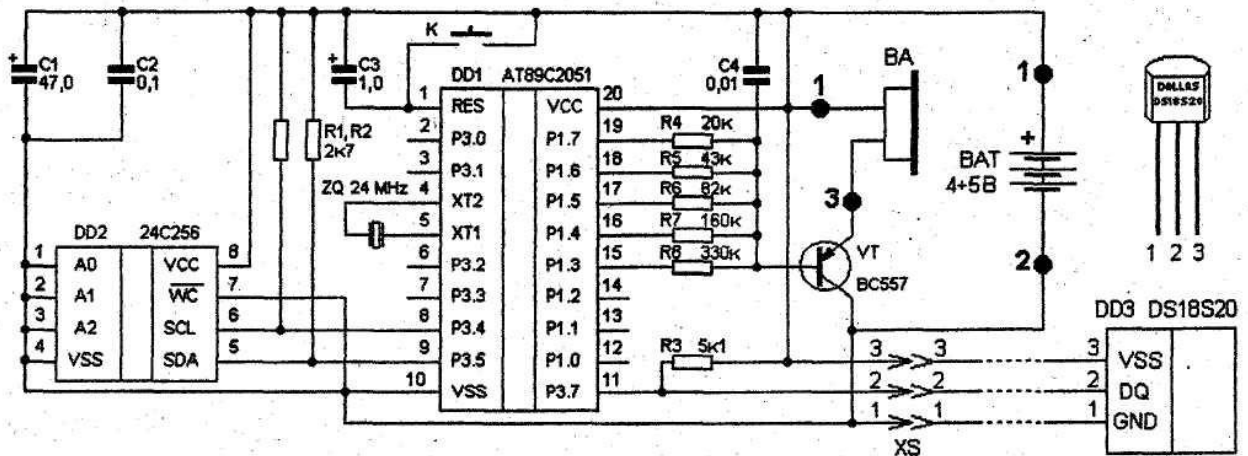


Рис. 4. Принципова схема пристрою

Мікросхема AT89C2051 являється центральним керуючим пристроєм, яка отримує живлення +4-5 В із джерела напруги встановленого у корпусі пристрою. У режимі виміру й «озвучення» температури, споживає струм не більше 40 мА (при напрузі живлення 5 В та опорі динаміка 8 Ом). Після «озвучення» прилад переходить в режим мікроспоживання та споживає струм не більше 3 мкА тому встановлення вимикача живлення не обов'язково. Датчиком температури використовується цифровий датчик температури DS18S20 з 1-провідним інтерфейсом (шина MicroLAN) фірми DALLAS. Датчик забезпечує точність виміру ± 0.5 градуса Цельсія. У приладі використовується крок виміру один градус. Межа вимірювання температур датчиком від -50 до +89 градусів Цельсія. Датчик може з'єднується із електронним блоком трьох-провідною лінією (загальний, інформаційний, та живлення) довжиною до 100 м. «Звукові» файли записані на ПК і «зжаті» стандартною програмою «Звукозапис» Windows. «Звукові» файли записані в енергонезалежну пам'ять DD2 24C256. Програма роботи записана у програмну пам'ять мікроконтролера DD1 AT89C2051. Після зчитування температури мікроконтролер озвучує її за допомогою послідовної вибірки із частотою 5,5 кГц значень амплітуди і установки її на резисторному цифрово-аналоговому перетворювачі R4-R8. Транзистор VT - підсилювач струму для низькоомного гучномовця BA.

Відтворення значення температури - звукове, наприклад, «плюс двадцять п'ять градусів Цельсія» або «мінус десять градусів Цельсія». У випадку несправності лінії зв'язку з датчиком термометр озвучує лише фразу «градусів Цельсія» без значення температури.

Для виміру й «озвучення» температури достатньо короткочасно натиснути кнопку керування. Після озвучування значення температури прилад переходить у режим мікроспоживання.

Пристрій може бути модифікований шляхом додавання інших датчиків температури, вибір потрібного буде здійснюватись за допомогою перемикача, для цього потрібно об'єднати спільні проводи та проводи живлення датчиків температури. Дана модифікація дозволить шляхом не великих витрат, отримувати інформацію про температуру в різних кімнатах, або ж за межами будинку.

Друковану плату для пристрою було розроблено в програмі Sprint Layout (рис. 5). Ключі мікроконтролера та мікросхеми пам'яті позначені отвором із квадратною окантовкою та встановлено в панельки для зручності. Друкована плата, кнопка керування, джерело напруги, гучномовець та датчик температури встановлено в пластмасовий корпус. Датчик температури винесено до краю корпусу для більшої точності вимірювання. Всі деталі на платі розташовані компактно, що дає змогу економити на загальному розмірі пристрою.

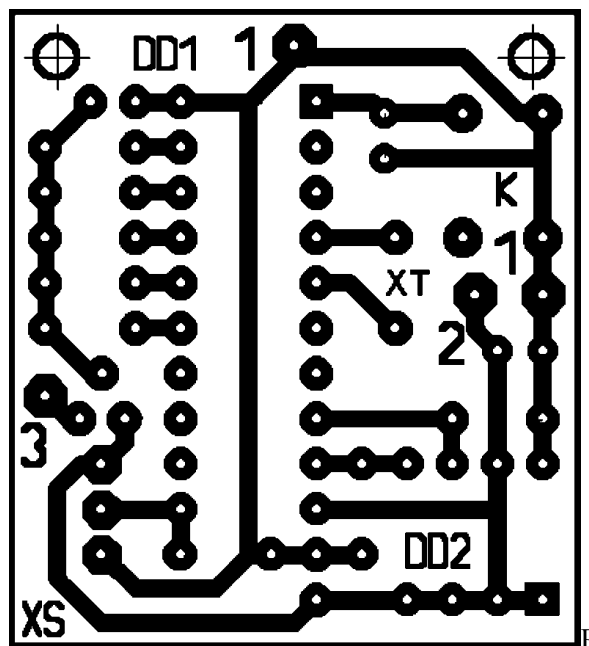


Рис. 5. Друкована плата пристрою

Висновки. Здійснено реалізацію програмно-апаратного комплексу на базі мікроконтролера AVR AT7294C. Проект повністю розроблений для вимірювання температури навколишнього середовища. Термометр може бути призначений для дистанційного вимірювання температури в різних місцях, наприклад в кімнаті та на вулиці. Для цього слід встановити датчики, об'єднавши їх спільні проводи та проводи живлення. Потрібний інформаційний провід перед вимірюванням температури можна вибрати за допомогою перемикача.

Термометр із голосовим повідомленням про значення температури знайде застосування в технологічних процесах, де потрібен слуховий контроль за температурою чи голосове повідомлення про значення температури. Пристрій є недорогим, компактным та може використовуватись як у побуті так і на підприємствах де потрібен контроль за температурою, а також стане в нагоді людям із вадами зору.

Список використаних джерел.

1. Вимірювальний прилад [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/вимірювальний_прилад (Дата звернення 01.03.17).
2. Цифрові вимірювальні прилади [Електронний ресурс]. – Режим доступу: <http://optictoday.com/katalog-statej/stati-na-ukrainskom/optichni-vimiryuvannya/cifrovi-vimiryuvalni-priladi.html> (Дата звернення 11.03.17).
3. Gerhart James Home Automation and Wiring//McGraw-Hill Professional, 31 March 1999. - ISBN 0-07-024674-2.
4. Harper et al, 2003. pp. 18–19
5. Mann William C. The state of the science//Smart technology for aging, disability and independence. - John Wiley and Sons, 7 July 2005. - ISBN 0-471-69694-3.
6. The best smart light you can buy [Електронний ресурс]. <http://www.theverge.com/2016/8/23/12560024/best-smart-lights-home-gadgets-philips-hue> (Дата звернення 25.03.17).
7. Михаил Гонтарь. Говорящий термометр [Журнал]. – Радиолюбби №1 2008 г
8. Говорящий термометр для дома [Електронний ресурс]. – Режим доступу: <http://www.troiti.com.ua/termometr.html> (Дата звернення 21.03.17).
9. Цифрові вимірювальні прилади [Електронний ресурс]. – Режим доступу: <http://optictoday.com/katalog-statej/stati-na-ukrainskom/optichni-vimiryuvannya/cifrovi-vimiryuvalni-priladi.html> (Дата звернення 02.03.17).
10. Микроконтролер AT89C2051 [Електронний ресурс]. – Режим доступу: <http://su0.ru/Mpao0> (Дата звернення 07.03.17).
11. Трамперт В. "Измерение, управление и регулирование с помощью AVR-микроконтроллеров" — МК-Пресс 2006г.
12. Микроконтролер AVR початок роботи [Електронний ресурс]. – Режим доступу: http://www.atmel.com/ru/ru/products/microcontrollers/avr/start_now.aspx (Дата звернення 01.04.17).

УДК 681.515.8

М.М. Поліщук, О.К. Каганюк, Р.В. Сопіжук
Луцький національний технічний університет

ДОСЛІДЖЕННЯ DC-AC ПЕРЕТВОРЮВАЧА З МІКРОКОНТРОЛЕРНИМ КЕРУВАННЯМ ЧАСТОТИ ІНВЕРТОРА

М.М. Поліщук, О.К. Каганюк., Р.В. Сопіжук. Дослідження DC-AC перетворювач з мікроконтролерним керуванням частоти інвертора. Досліджено перетворювач електричної енергії, його режими роботи та основні складові частин від яких залежить потужність роботи пристрою. Досліджено режими роботи та характеристики інвертора при різних елементах пристрою, побудовані графіки роботи перетворювача, дано рекомендації по оптимізації, вказано переваги та недоліки проєктованого рішення.

Ключові слова: інвертор, мікроконтролер, ATmega328P, план Бокса-Бенкіна, трансформатор, транзистори, частота.

Н.Н. Полищук, А.К. Каганюк, Р.В. Сопижук. Исследование DC-AC преобразователь микроконтроллерным управлением частоты инвертора. Исследованы преобразователь электрической энергии, его режимы работы и основные составляющие частей от которых зависит мощность работы устройства. Исследованы режимы работы и характеристики инвертора при различных элементах устройства, построенные графики работы преобразователя, даны рекомендации по оптимизации, указано преимущества и недостатки проектируемого решения.

Ключевые слова: инвертор, микроконтроллер, ATmega328P, план Бокса-Бенкина, трансформатор, транзисторы, частота.

M.M.Polishchuk, A.K.Kaganiuk., R.V.Sopizhuk. Research DC-AC converter with microcontroller-controlled frequency inverter. Researched electrical energy converter, its modes of operation and the main components of which depend on the capacity of the device. Investigated modes and features inverter with various elements of the device, the graphs of the converter is given recommendations on optimization given the advantages and disadvantages of the designed solution.

Keywords: inverter microcontroller, ATmega328P, plan Box-Banking, transformer, transistor, frequency.

Вступ. Перетворювачем електричної енергії є пристрій, який пов'язує дві або більше електричні системи з відмінними один від одного параметрами і дозволяє по заданому закону змінювати ці параметри забезпечуючи обмін електричною енергією між зв'язуваними системами [1]. Напівпровідникові перетворювачі напруги - електронні трансформатори, що зв'язують системи змінного і постійного струму, а саме перетворювачі постійної напруги в змінну - інвертори.

Історично першими виникли перетворювачі енергії, на базі електромеханічних пристроїв. Електричний двигун перетворював електричну енергію в механічну, яка за допомогою генератора перетворювалась в електричну з потрібними параметрами [2]. Сучасні пристрої перетворення енергії не містять рухомих елементів, тому їх називають статичними перетворювачами, зокрема інвертори DC-AC перетворюють, постійну напругу 12 В у змінну напругу 220 В. Інвертор значно дешевший за міні-електростанцію, є мініатюрним та легким [3]. Спільно з одним, або декількома акумуляторами він може працювати як автономне джерело безперебійного живлення для будинку, котельної, пожежних і охоронних систем. Якщо є мережева напруга 220 Вольт, він просто пропускає його "крізь" себе і, при необхідності, заряджає акумулятори. Якщо напруга в мережі зникла, інвертор миттєво починає генерувати змінну напругу 220 Вольт від акумуляторів [4]. Час автономної роботи залежить від потужності навантаження і ємності акумуляторів. Так, наприклад, чотирьох акумуляторів по 190 А/г вистачить приблизно на 16 годин автономної роботи при постійному навантаженні 0,5 КВт [5]. При появі мережевої напруги прилад автоматично перемкнеться в початковий стан очікування і зарядить акумулятори.

Мета та завдання розробки. Мета дослідження – дослідження DC-AC перетворювача з мікроконтролерним керуванням частоти інвертора. Дане дослідження дозволить автоматизувати і удосконалити деякі елементи пристрою, а також, провести аналіз вхідних та вихідних даних.

Завдання дослідження зумовлюються метою роботи і полягають у наступному:

- проаналізувати вхідні та вихідні дані DC-AC перетворювача;
- дослідити елементну базу інвертора;
- виявити та систематизувати фактори що впливають на вихідні дані пристрою;
- здійснити оптимізацію елементної бази DC-AC перетворювача для кращого функціонування;

Аналіз існуючих перетворювачів напруг. Найпопулярніші технічними рішеннями для інверторів DC-AC стали класичні схеми мостових інверторів та їх модифікації. Наприклад, функціональну схему інвертора серії DA, що перетворює нестабільну напругу постійного струму в

стабілізовану (в тому числі і по частоті) однофазну напругу змінного струму, наведено в [6]. На вході силового двоступеневого фільтра, каскад формує багато-імпульсні послідовності. Силовий трансформатор, що входить до складу перетворювача постійної напруги живиться від мостового каскаду транзисторів MOSFET. При такому варіанті побудови інвертора, незважаючи на додаткове перетворення енергії в імпульсному перетворювачі напруги (440 В), при раціональному проектуванні перетворювача постійної напруги (в даному випадку робоча частота дорівнює 80 кГц), досягаються високі енергетичні і масо-габаритні характеристики.

У промисловому виконанні найбільшого поширення набули двотактні мостові схеми виконані на польових транзисторах фірми International Rectifier IRFP23N50L. Мостовий каскад і силовий фільтр охоплені негативним зворотним зв'язком, що дозволяє забезпечити високі енергетичні характеристики інвертора. Для забезпечення необхідної форми синусоїдальної вихідної напруги використовується дволанковий LC-фільтр змінного струму, який виконує функції дроселя виконаного на кільцевому сердечнику з МО - пермалой.

Однією із сфер застосувань інверторів є альтернативна енергетика. Наприклад, американська фірма Texas Instruments пропонує рішення C2000™ Solar DC/AC SinglePhase Inverter на базі своїх нових мікроконтролерів C2000™ F28M35H52C. У пристрої конструктивно реалізовано лише одна фаза змінної напруги. В основі інверторного каскаду перетворення лежить повний міст DC/AC. Цифровий контроль дозволяє реалізувати інтерфейси віддаленого моніторингу та контролю пристрою. Сумарний коефіцієнт нелінійних спотворень (THD) менший ніж на 5 відсотків, а повний ККД навантаження більший ніж 96%. Єдиним мінусом даного інверторного каскаду є висока ціна, яка досягає 450\$. [7].

До світових лідерів в галузі силової електроніки входить також американська компанія Xantrex [8]. Компанія була заснована більше 20-ти років тому і в результаті злиття виробників інверторів перетворилася на великий промисловий холдинг. Продукцію компанії відрізняє висока відмовостійкість, в конструктиві вже закладено грозовий захист, захист від екстрених перевантажень, захист від некоректної інсталяції. Останні 10-років компанія активно розвиває напрямок керуючих систем для вітрогенераторів, сонячних батарей і автоматизації систем автономного живлення. Ціна на продукцію Xantrex дуже висока, але в деяких випадках (універсальні контролери для сонячних батарей) конкурентні альтернативи відсутні.

Одним із лідерів азіатського ринку є компанія CyberPower Systems [9]. Це тайванський виробник ББЖ та інверторів. Його продукція відрізняється відповідністю заявленим характеристикам, чистим синусом на виході інверторів та високою довговічністю.

Щодо елементної бази, то безперечним лідером виробником мікросхем обов'язки залишається компанія MAXIM™. У каталозі продукції що випускається, широко застосовуються перетворювачі DC-AC на базі ШІМ-регулювання (наприклад серії MAX 1739, 1839) для регулювання яскравості флуоресцентних ламп [10].

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Наукова новизна дослідження полягає в наступному:

- досліджено вплив геометричних параметрів вторинної обмотки трансформатора на вихідну потужність перетворювача напруги.
- проведено лабораторні випробування розробленого пристрою для перетворення напруги із застосуванням математичного методу планування експерименту по чотирьох факторах на трьох рівнях, а саме симетричного некомпозиційного плану реалізації експерименту Бокса-Бенкіна другого порядку.
- досліджено вплив вхідної напруги, вихідної частоти, потужності транзисторів, геометричних параметрів вторинної обмотки трансформатора на вихідну потужність DC-AC інвертора.

Для досягнення поставленого завдання було проведено аналіз вихідних даних при різних параметрах вхідних напруг, порівняльні характеристики можливостей перетворювача напруги із зміненими елементами.

Дослідження DC-AC перетворювача проводилась методом експерименту, оскільки цей метод найбільш точніше продемонструє роботу інвертора за різних умов та елементів цього пристрою. У процесі дослідження будуть отримані дані, які характеризують умови роботи інвертора в оптимальному режимі, що наочно підтверджується графічним матеріалом. при різних режимах роботи інвертора. Лише на основі перевірки адекватності моделі формувалася найбільш оптимальна та ефективний режим роботи пристрою. Запропоновані зміни у перетворювача напруги дозволять пришвидшити роботу та збільшити потужність даного пристрою, завдяки

заміні параметрів трансформатора, транзистора та мікроконтролера, що у свою чергу дає можливість інвертору підключати більше електричних пристроїв та стабілізує вихідну напругу, а саме модифіковану синусоїду.

Результати дослідження показали працездатність інвертора з раціональними параметрами визначивши при цьому найоптимальнішу та найефективнішу елементну базу для нього [11]. Також був проведений аналіз роботи пристрою для різних входних напруг. Структурна схема пристрою зображена на рис. 1.

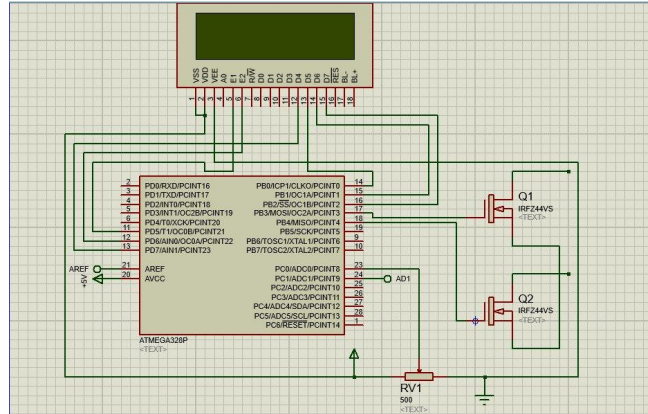


Рис. 1. Модель перетворювача у програмі Proteus

Джерело енергії постійного струму, в найпоширенішому випадку акумулятор 12В, підключається до трансформатора через трипозиційний комутатор. Комутатор є набором електронних ключів, що забезпечує 3 стани: до первинної обмотки трансформатора підключено джерело живлення позитивної полярності, до первинної обмотки трансформатора підключене джерело живлення негативної полярності та стан коли первинна обмотка короткозамкнена. Послідовно перемикаючи ці стани, на первинній обмотці формується змінна напруга частотою 50Гц і амплітудою 12В. На вторинній обмотці трансформатора при цьому формується напруга з тією ж частотою і формою, проте ефективна напруга складає 220В. Вихідна напруга знімається з вторинної обмотки, тому має аналогічні параметри.

На рис. 1 зображено блок-схему роботи пристрою, алгоритм роботи якого полягає в почерговому вмиканні і вимиканні 17 і 18 пінів мікроконтролера ATmega 328P. Відповідні сигнали надходять на затвори польових транзисторів, які в свою чергу керують напругою і струмом на трансформаторі, що піднімає напругу з 12 В до 220 В.

Наприклад, інвертор, розрахований на напругу джерела 12В, працює від розрядженого акумулятора з напругою 10В. При цьому амплітуда напруги на виході знижується пропорційно до 259В. Схема управління змінює коефіцієнт заповнення вихідної напруги до 0.72, при цьому ефективна напруга залишається рівною 220В. Проте форма напруги та її амплітуда змінюється, а це може бути недопустимо для деяких навантажень. Фото пристрою представлено на рис. 2.



Рис. 2. Зовнішній вигляд розробленого DC-АСперетворювача

Методика лабораторно-виробничих випробувань пристрою. Складність технологічного процесу перетворювача напруги при його структурі не дозволяє повністю визначити раціональні конструктивні параметри та оптимальні режими роботи пристрою аналітичним шляхом. Тому, для

встановлення впливу на потужність та стабільність роботи пристрою елементів було проведено лабораторно-виробничі випробування розробленого пристрою при різних складових елементах із застосуванням математичного методу планування експерименту.

Аналіз результатів теоретичного дослідження процесу оптимізації параметрів інвертора, показав що значний вплив на вихідну напругу та потужність пристрою має визначена потужність польових транзисторів та товщина провідника вторинної обмотки трансформатора та вхідна напруга. Результати експериментальних досліджень показують, що на потужність також впливає вихідна частота перетворювача. Тому, для дослідження роботи пристрою було вирішено провести чотирих-факторний експеримент.

Для проведення повнофакторного експерименту необхідно провести дослідів, які визначаються за формулою:

$$N = m^k, \quad (1)$$

де m – кількість рівнів дослідження;

k – кількість факторів у серії дослідження.

Проведення повного факторного експерименту по чотирьох факторах на трьох рівнях вимагає проведення 81 дослідів. Враховуючи трудомісткість та вартість проведення дослідів, важливо було зменшити їх кількість до оптимально-необхідного значення. Цю задачу дозволяє розв'язати використання симетричного некомпозиційного плану реалізації експерименту Бокса-Бенкіна другого порядку. Для проведення чотири-факторного експерименту, згідно з вищезгаданим планом, необхідно провести 27 дослідів.

Планування експериментального дослідження включало такі етапи:

- кодування факторів;
- складання таблиці факторів і рівнів варіювання;
- складання матриці планування;
- реалізація плану експерименту згідно з матрицею планування;
- складання рівняння регресії для визначення коефіцієнтів;
- оцінка значущості коефіцієнтів регресії;
- перевірка адекватності отриманої математичної моделі.

При складанні таблиці факторів і рівнів варіювання (табл.1.1) враховували результати попередніх досліджень. План Бокса-Бенкіна розрахований на використання трьох рівнів для кожного фактора: верхнього (+1), нульового (0) і нижнього (-1) [12].

Таблиця 1.1. Фактори і рівні варіювання

Рівні варіювання	Фактори			
	Вхідна напруга $U_{вх}$, В	Частота вихідна V , Гц.	Потужність польових транзисторів P , Вт	Площа січення вторинної обмотки трансформатора S , мм ²
	x_1	x_2	x_3	x_4
Верхній (+1)	15	150	300	0,7
Нульовий (0)	12	100	200	0,5
Нижній (-1)	9	50	100	0,3
Інтервал варіювання, ε	3	50	100	0,2

Дослідження проводили, використовуючи матрицю планування. У розкодованому вигляді матриця планування експерименту і порядок проведення дослідів представлені у таблиці. Зв'язок між кодованими і натуральними значеннями факторів встановлювався залежностями:

$$x_1 = \frac{U_{\text{вх}} - U_{\text{вх}0}}{\varepsilon_1}; x_2 = \frac{v - v_0}{\varepsilon_2}; x_3 = \frac{P - P_0}{\varepsilon_3}, x_4 = \frac{S - S_0}{\varepsilon_4} \quad (2)$$

де x_1, x_2, x_3, x_4 – кодовані значення факторів, відповідно, вхідна напруга, вихідна частота, потужність польових транзисторів та товщина провідника вторинної обмотки трансформатора;

$U_{\text{вх}}, v, P, S$ – натуральні значення факторів, відповідно, вхідна напруга, вихідна частота, потужність польових транзисторів та товщина провідника вторинної обмотки трансформатора;

$U_{\text{вх}0}, v_0, P_0, S_0$ – натуральні значення факторів на нульовому рівні;

$\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ – інтервали варіювання факторів.

Нелінійний характер очікування функцій відгуку ступені дослідження інвертора в області факторного простору поданий у вигляді рівняння регресії:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + b_{12}x_1x_2 + b_{13}x_1x_3 + b_{14}x_1x_4 + b_{23}x_2x_3 + b_{24}x_2x_4 + b_{34}x_3x_4 + b_{11}x_1^2 + b_{22}x_2^2 + b_{33}x_3^2 + b_{44}x_4^2 \quad (3)$$

Коефіцієнти регресії після реалізації плану експерименту визначено за такими формулами:

$$b_0 = 0,33333 \sum_{j=1}^n y_j - 0,16667 \sum_{i=1}^k \sum_{j=1}^n x_{ij}^2 y_j, \quad (4)$$

$$b_i = 0,08333 \sum_{j=1}^n x_{ij} y_j, \quad (5)$$

$$b_{ii} = 0,125 \sum_{j=1}^n x_{ij}^2 y_j + 0,0625 \sum_{i=1}^k \sum_{j=1}^n x_{ij}^2 y_j - 0,16667 \sum_{j=1}^n y_j, \quad (6)$$

$$b_{ir} = 0,25 \sum_{j=1}^n x_{ij} x_{rj} y_j, \quad (7)$$

де j - номер дослідів в матриці планування;

n - кількість дослідів в матриці планування;

y_j - значення функції відгуку в j -му досліді;

k -кількість факторів;

x_{ij}, x_{rj} - кодовані значення i -го чи r -го фактору в j -му досліді;

i, r - номери факторів.

Дисперсії коефіцієнтів регресії та їх коваріації визначено за формулами:

$$S_{b_0}^2 = 0,33333 S_y^2, \quad (8)$$

$$S_{b_i}^2 = 0,08333 S_y^2, \quad (9)$$

$$S_{b_{ii}}^2 = 0,1875 S_y^2, \quad (10)$$

$$S_{b_{ir}}^2 = 0,25 S_y^2, \quad (11)$$

$$\text{cov}_{b_0 b_{ii}} = -0,16667 S_y^2, \quad (12)$$

$$\text{cov}_{b_{ii} b_{rr}} = 0,0625 S_y^2. \quad (13)$$

Значимість коефіцієнтів регресії перевіряли шляхом порівнювання абсолютної величини цих коефіцієнтів з їх довірчими інтервалами. Довірчі інтервали розраховували згідно з формулами:

$$\Delta b_0 = t_{0,05;f_1} S_{b_0}, \quad (14)$$

$$\Delta b_i = t_{0,05;f_1} S_{b_i}, \quad (15)$$

$$\Delta b_{ir} = t_{0,05;f_1} S_{b_{ir}}, \quad (16)$$

$$\Delta b_{ii} = t_{0,05;f_1} S_{b_{ii}}, \quad (17)$$

де $t_{0,05;f_1}$ - табличне значення критерію Ст'юдента при 5%-му рівні значущості та $f_1 = n_0 - 1$ - числа ступенів вільності дисперсії відтворюваності (n_0 - кількість дослідів у центрі плану).

Коефіцієнт регресії вважали статистично значущим, коли його абсолютна величина більша довірчого інтервалу або рівна йому. Незначущі коефіцієнти видаляли із моделі.

Перевірку адекватності рівняння проводили за допомогою F_Φ - критерію Фішера. Адекватність отриманої моделі матиме місце, якщо розрахункове значення критерію $F_{f_2;f_1}^{\text{розр.}}$ менше табличного для прийнятого рівня значимості:

$$F_{f_2;f_1}^{\text{розр.}} \leq F_{0,05;f_2;f_1}^{\text{табл.}}, \quad (18)$$

де $F_{0,05;f_2;f_1}^{\text{табл.}}$ - табличне значення критерію Фішера при 5%-му рівні значущості і ступенях вільності дисперсії неадекватності f_2 та дисперсії відтворюваності f_1 .

$f_2 = n - k'$ - число ступенів вільності дисперсії неадекватності з урахуванням числа k' залишених коефіцієнтів регресії (у тому числі і b_0).

Розрахункове значення критерію Фішера становить:

$$F_{f_2;f_1}^{\text{розр.}} = \frac{S_{\text{неад.}}^2}{S_y^2}, \quad (19)$$

де $S_{\text{неад.}}^2$ - дисперсія неадекватності, яка визначена з f_2 - числом ступенів вільності;

S_y^2 - дисперсія відтворюваності експерименту, яка визначена з f_1 - числом ступенів вільності.

Згідно з матрицею планування експерименту, кількість дослідів на нульовому рівні рівна трьом. Це дозволило використати для визначення дисперсії відтворюваності експерименту таку формулу:

$$S_y^2 = \frac{\sum_{u=1}^{n_0} (y_{0_u} - \bar{y}_0)^2}{f_1}, \quad (20)$$

де u - номер дослідів у центрі плану;

y_{0_u} - значення функції відгуку в u -му досліді в центрі плану;

$\bar{y}_0$ - середнє арифметичне значення функції відгуку отримане за результатами n_0 дослідів у центрі плану;

Дисперсію неадекватності визначали, користуючись наступною формулою:

$$S_{\text{неад.}}^2 = \frac{SS_{\text{неад.}}}{f_2}. \quad (21)$$

При розрахунку дисперсії $S_{\text{неад.}}^2$ враховували ту обставину, що при проведенні експериментів з усіх дослідів плану повторювався тільки один (дослід в центрі плану). Тому, сума квадратів $SS_{\text{неад.}}$ дорівнює:

$$SS_{\text{неад.}} = n_0(\hat{y}_0 - \bar{y}_0)^2 + \sum_{j=1}^{n-n_0} (\hat{y}_j - y_j)^2, \quad (22)$$

де $\hat{y}_0$ - значення функції відгуку в досліді із центру плану, що розраховане за рівнянням регресії;

$\hat{y}_j$ - значення функції відгуку в j-му досліді, що розраховане за рівнянням регресії;

y_j - значення функції відгуку в j-му досліді, що визначене експериментально[12].

Висновки. Використана методика проведення експериментів планом Бокса-Бенкіна дозволить:

1. Отримати математичну модель процесу оптимізації перетворювача напруги при його зміні елементної бази у вигляді рівняння регресії, функцією відгуку якого є потужність інвертора;
2. Провести аналіз вихідних даних при різних вхідних напругах, порівняльні характеристики можливостей перетворювача напруги із зміненими елементами;
3. Отримані результати проведених досліджень можуть вказати на певні особливості зміни властивостей інвертора. Так за збільшенням потужності польових транзисторів, відбувається збільшення потужності перетворювача. Крім цього, за наявності більшого провідника вторинної обмотки, що необхідно враховувати при виборі трансформатора, теж збільшується потужність. У випадку збільшення частоти пристрою можуть нести негативну дію на ефективність транзисторів та мікроконтролера. Тому, при конструюванні оптимальних складових пристрою необхідно в його конструкції застосовувати мультівібратор.

1. Крогерис А., Рашевіц К., Рутманіс Л. и др. Полупроводниковые преобразователи электрической энергии / Под ред. А. Крогериса. Рига: Зинатне. 1969.
2. Ирвинг М., Готтлиб. Источники питания. Инверторы, конверторы, линейные и импульсные стабилизаторы. – 2-е изд. – М.: Постмаркет, 2002. – 544 с.
3. Преобразователи напряжения и автомобильные инверторы напряжения (12-220В, 220-12В, 24-12В) [Електронний ресурс]. – Режим доступу: <http://auto-line.net/ru/shop/car-electrics/car-voltage-converters>.
4. «Польовий» перетворювач напруги [Електронний ресурс]. – Режим доступу: <http://www.radiosvoboda.org/content/article/26936954.html>. (Дата звернення 01.04.17)
5. Схема преобразователя 12\220 [Електронний ресурс]. – Режим доступу: <http://monitor.espec.ws/section44/printview127136.html>.
6. Транзисторные DC-AC преобразователи напряжения: характеристики, структурные схемы, рекомендации по применению // "Силовая электроника" № 2, 2004.
7. C2000™ Solar DC/AC Single Phase Inverter // <http://www.ti.com/tool/TIDM-SOLAR-ONEPHINV#descriptionArea>
8. Power inverters [Електронний ресурс]. – Режим доступу: <http://www.xantrex.com/power-products/power-inverters/overview.aspx> (дата звернення 05.04.17)
9. Product literature [Електронний ресурс]. – Режим доступу: [https://www.cyberpowersystems.com/support/knowledge-center/general-literature\(дата звернення 05.04.17\)](https://www.cyberpowersystems.com/support/knowledge-center/general-literature(дата звернення 05.04.17))
10. WideBrightnessRangeCCFLBacklightControllers [Електронний ресурс]. – Режим доступу: <https://datasheets.maximintegrated.com/en/ds/MAX1739-MAX1839.pdf> (дата звернення 12.04.17).
11. Здолбіцька Н.В. DC-AC перетворювач з мікроконтролерним керуванням частоти інвертора / Н.В. Здолбіцька, А.П. Здолбіцький, Р.В. Сопіжук, В.В. Супрунук // Науковий журнал "Комп'ютерно-інтегровані технології: освіта, наука, виробництво". – Луцьк: РРВ Луцького НТУ. – Вип. 23. – 2016. – С. 25-30.
12. Аністратенко В.О., Федоров В.Г. / Математичне планування експериментів в АПК: Навч. посібник. -К.Вишашк., 1993.-375 с.

УДК 004.31

М.М. Поліщук, С.В. Гринюк, В.В. Дмитерчук
Луцький національний технічний університет

РОБОТОТЕХНІЧНА СИСТЕМА АВТОМАТИЧНОГО ПОШУКУ ДЖЕРЕЛА СВІТЛА

М.М. Поліщук, С.В. Гринюк, В.В. Дмитерчук. Робототехнічна система автоматичного пошуку джерела світла. У статті представлено огляд ринку робототехніки, обґрунтування вибору елементної бази, а також пропонується бюджетний пристрій автоматичного пошуку світла з можливістю модернізування для будь-яких потреб.

Ключові слова: робот, система, датчик, PTC, Arduino, Atmega 328, сонячна панель.

М.М. Полищук, С.В. Гринюк, В.В. Дмитерчук. Робототехническая система автоматизированного поиска источника света. В статье представлен обзор рынка робототехники, обоснование выбора элементной базы, а также предлагается бюджетное Устройство автоматического поиска света с возможностью модернизации для любых нужд.

Ключевые слова: робот, система, датчик, PTC, Arduino, Atmega 328, солнечная панель.

M.M. Polishchuk, S.V. Grunyk, V.V. Dmyterchuk. The robotics system of automatic search of source of light. An overview of the market robotics study and selection of components, as well as the proposed budget device automatically light with a search possibility of upgrading for any needs.

Keywords: robot, system, sensor, RTS, Arduino, Atmega 328 solar panels.

Постановка проблеми. Сучасною промисловістю виробляється велика кількість різноманітних автоматичних роботів, проте переважна більшість має обмежену функціональність або велику вартість[1]:

–робот Wow Wee MINI призначений для розваг та розвитку дітей має обмежений набір команд, не оснащений автономністю, має малу прохідність і може працювати лише декілька годин після чого його потрібно відправляти на підзарядку [2];

–робот – порохотяг Xiaomi Mi Robot Vacuum (SKV4000CN) – автоматичний пилесос має датчики пилу та лазерні датчики відстані, електронний компас, гіроскоп, датчик падіння, шпідометр та багато іншого, про те, він має високу ціну та лиш одну функцію — прибирання підлоги [3];

–RiSE – шестиногий робот сконструйований Boston Dynamics у співпраці з Пенсильванським університетом, Університетом Карнегі – Меллон, Каліфорнійським університетом у Берклі, Стенфордським університетом та Коледжем Льюїса і Кларка., здатний підніматися вертикальними поверхнями: стіни, дерева, огорожі. Для пересування використовує ноги з мікрогачками та хвіст, може змінювати позу відповідно до кривизни поверхні. RiSE має довжину 0.25 м та масу 2 кг, Швидкість пересування — 0,3 м/с. Кожна нога робота керована двома електромоторами. Бортовий комп'ютер керує ногами, забезпечує зв'язок та обробку даних сенсорів. Але робот має велику вартість складання, тому випущений в єдиному екземплярі і поки тестується [4].

Актуальність. Розробка бюджетного автоматичного пристрою пошуку світла з можливістю модернізування для будь-яких потреб є актуальною, оскільки на ринку не представлені такі системи, а їхню нішу заповнюють аматорські розробки, які доволі часто є не доопрацьованими та із закритим кодом, що ускладнює їх подальший розвиток та модернізацію.

Аналіз останніх досліджень і публікацій. Автоматизація та робототехніка є одним із пріоритетних напрямків розвитку промисловості в усіх розвинених країнах. Сучасними попередниками роботів є різного роду пристрої для маніпулювання на відстані об'єктами, безпосередній контакт людини з якими небезпечний або неможливий. Дана галузь є важливою також і в міжнародних програмах розвитку науково-технічного потенціалу. Перші пристрої такого роду були пасивними, тобто механізмами без приводів, і служили для повторення на відстані рухів руки людини цілком за рахунок його мускульної сили. Потім були створені маніпулятори із приводами й керовані людиною різними способами аж до біоелектричного[5].

В наш час роботи усе частіше застосовуються в широкому спектрі галузей, таких як: побутова (домашня) робототехніка, промислова робототехніка, медична робототехніка, воєнна робототехніка, планетарна робототехніка, персональна робототехніка, еволюційна робототехніка, біометрична робототехніка, біологічна робототехніка, мікроробототехніка, наноробототехніка, нейроробототехніка [6].

На світовому ринку робототехніки впевнено лідирують Японія та Німеччина. Вони виготовляють більше 50% всієї роботизованої продукції в світі. Говорити про якісь серйозні досягнення України в цій області поки рано. Про те тут стрімко розвивається навчальний сегмент як приватні курси робототехніки для дорослих та дітей так і безплатні гуртки, наприклад: «Винахідник» (Київ, Бровари, Буча, Дніпро, Львів, Івано-Франківськ, Вінниця, Харків, Одеса), RoboCode (Київ, Ірпінь), RobotSchool (Київ, Полтава), RoboUA (Київ), Robo.House (Київ), Funtronica (Одеса) та інші.

Компанія "Стандарт-ПАК" розробляє самохідні платформи, роботизовані вози, автоматичні навантажувачі та робобуксири. Роботи можуть переміщувати вантажі, оснащені датчиками руху та переміщуються зі швидкістю від 0,5 м/с. Це перші в світі аналоги Kiva від Amazon Fetch, Freight від Fetch Robotics та OTTO 100 від Kitchener, які пропонуються в продажі для логістичних центрів України та Європи. Споживачі таких роботів – логістика і виробництво де потрібна автоматизація переміщення (торгові та розважальні центри, сільське господарство, банки).

В Drone.UA створюють продукти в області безпілотних технологій. Компанія веде діяльність в аграрній сфері, енергетиці та нафтогазовій промисловості, а також в сферах геодезії та топографії. Їх технології використовуються на більш чим 2 млн гектарів посівних площ України. Компанія випускає більше десяти видів мультироторних платформ (ціни стартують від 20 тис. грн) та літальна платформа PD1900. Ця модель призначена для отримання даних аерофотозйомки і здатна покрити до 400 га площі за 100 хвилин. Його вартість складає 67 тис. грн.

Концерн ДК "Укроборонпром" об'єднує декілька підприємств розробників, які конструюють бойову роботизовану техніку. Один з учасників концерну, "СпецТехноЕкспорт", побудував перший в Україні безпілотний БТР "Фантом" – це 6-колісний робот, здатний перевозити бойові комплекти, транспортувати поранених та виконувати інші задачі. Робот може рухатись на відстань до 20 км. Керування виконується по захищеному радіоканалу або за допомогою волоконного кабелю.

Також в "Укроборонпром" будують безпілотні літальні апарати. Остання розробка — БПЛА ANSER, розроблений учасником концерну НДП "Спайтек". Безпілотник може знаходитись в польоті від 6 до 12 годин, піднімати вагу до 5 кг та використовувати зашифровані канали передачі даних.

В IT – компанії "ELEKS" також розробляють роботизовані продукти. Безпілотний танк ET-1 (Танкетка), розроблений для забезпечення безпеки українських солдат. З його допомогою можна евакуйовувати поранених, вести розвідку в місцях не доступних людині. Його також можна застосувати як вантажний автомобіль чи як трактор в сільському господарстві.

В запорізькій компанії "Інфоком ЛТД" розробляють безпілотний автомобіль. БНТС оснащується системою сенсорів та датчиків, щоб пристрій мав машинний зір і міг автономно приймати рішення або передавати інформацію оператору, який керує машиною дистанційно. Вже є 3 прототипи на дистанційному керуванні. Зараз йде робота по реалізації функціоналу автономних дій автомобіля [7].

Отже, більшість існуючих на ринку роботів мають або дуже обмежений функціонал, або дуже велику вартість. Розроблена робототехнічна система дозволить за мінімальну кількість коштів отримати автоматизований пристрій, що забезпечується завдяки використанню платформи Arduino, поширеної компонентної бази, та вільного ПЗ. Ліцензування розробленого комплексу за допомогою відкритої ліцензії дозволить будь-кому додати потрібний функціонал чи вдосконалити існуючий.

Виклад основного матеріалу. Робототехнічна система складається з таких основних блоків (рис. 1): керувальний пристрій, ключі для моторів та мотори, ультразвуковий датчик та сервопривід, джерело живлення, імпульсний перетворювач, плата заряду, сонячна панель.

В якості керувального пристрою було обрано платформу Arduino Uno R3 CH340, як недорогий та поширений контролер з достатньою швидкодією. Arduino має вбудований стабілізатор напруги на 5 В, тому може бути заживленою від одного джерела живлення із джерелом світла. Джерелом живлення виступають два літєвих елементи 18650 ввімкнені послідовно. Їхня напруга складає 8.4 В, тому було використано понижувальний імпульсний перетворювач LM2596 Mini 360 DC. Його характеристики показано в таблиці 1.

Джерело живлення також заряджається від сонячних панелей. Для оптимальної роботи системи було обрано сонячні панелі розміру 110*69мм. Робоча напруга 5 В, 250 мА. Оскільки зарядка літєвих акумуляторів послідовно ввімкнених без балансування небезпечно — елементи

заряджаються роздільно, кожна від своєї плати заряду (рис. 2). Також літєві елементи можна заряджати через ці ж плати заряду від будь-якого джерела на 5В 1А, наприклад USB.



Рис. 1. Структурна схема робототехнічної системи

Табл. 1. Основні характеристики імпульсного перетворювача LM2596 Mini 360 DC

Максимальна вхідна напруга	4,75 В
Мінімальна вхідна напруга	23 В
Максимальна вихідна напруга	1 В
Мінімальна вихідна напруга	17 В
Максимальний довготривалий вихідний струм	1,8 А
Максимальний імпульсний вихідний струм	3 А
Максимальний коефіцієнт корисної дії	96%
Робоча частота	340 КГц
Пульсації	30 мВ
Діапазон робочих температур	-40°C - +85°C

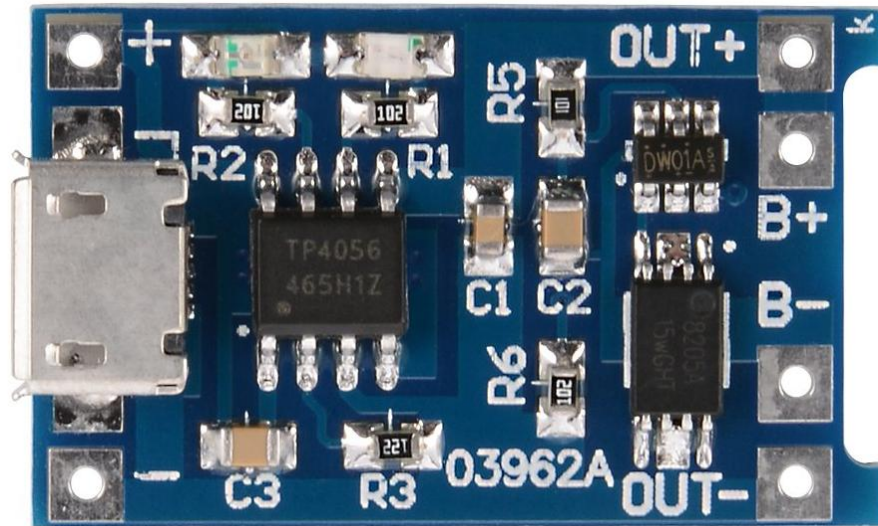


Рис. 2. Плата зарядки для акумулятора 18650

Arduino подає ШІМ – сигнал ключам, які комутують мотори, що дозволяє регулювати обороти моторів постійного струму. Для ключів використовуються польові N-канальні транзистори, що дозволяє підібрати необхідні параметри під джерела світла та комутувати навантаження, які живляться більшою напругою, ніж логічний рівень контролера.

Фрагмент коду керування двигунами постійного струму:

```
void direction(uint8_t dir) {  
  switch(dir) {  
    case FORWARD:  
      analogWrite(MOTORLEFT, MOTORLEFTMAX);  
      analogWrite(MOTORRIGHT, MOTORRIGHTMAX);  
      break;  
    case RIGHT:  
      analogWrite(MOTORLEFT, MOTORLEFTMAX);  
      analogWrite(MOTORRIGHT, 0);  
      break;  
    case LEFT:  
      analogWrite(MOTORLEFT, 0);  
      analogWrite(MOTORRIGHT, MOTORRIGHTMAX);  
      break;  
    default:  
      analogWrite(MOTORLEFT, 0);  
      analogWrite(MOTORRIGHT, 0);  
      break;  
  }  
}
```

Для оминання перешкод використовується ультразвуковий датчик, який закріплений на сервоприводі. В проекті було використано сервопривід моделі Tower Pro 9g SG90 (рис. 3), робоча напруга якого від 3В до 7,2В.



Рис. 3. Сервопривід Tower Pro 9g SG90: 1 – коричневий провід — земля (Ground, підключається до піну GND на платі Arduino); 2 – червоний провід — живлення +5V (підключається до піну 5V на платі Arduino); 3 – жовтий провід — сигнал керування (підключаємо до цифрового піну Arduino)[8].

Схема підключення сервоприводу до Arduino представлена на рис. 4.

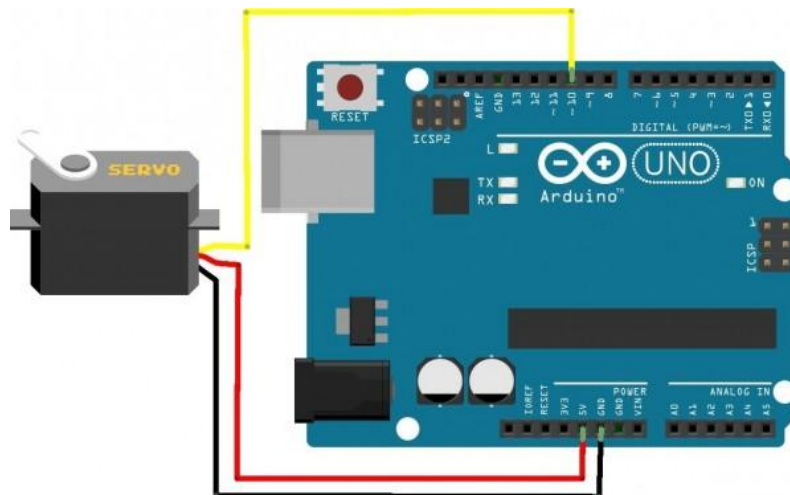


Рис. 4. Схема підключення сервоприводу до Arduino

Фрагмент коду роботи сервоприводу:

```
// dir -90...+90
void headdir(int8_t dir) {
  head.write(dir + 90 + HEADCORRECT);
}
```

Ультразвуковий датчик слугує для визначення відстані до перешкоди. Модуль формує запускаючий імпульс та вимірює значення ехо-сигналу:

$$l = \frac{t \cdot S}{2}$$

де l – відстань до перешкоди;
 S – швидкість звуку (340 м/с);

t – час за який імпульс подолає шлях від датчика до перешкоди і назад.

В проекті було використано ультразвуковий датчик моделі HC-SR04. Датчик має робочу напругу 5В та споживає 2мА в стані спокою і 15мА під час роботи, відповідно. Діапазон роботоzдатної відстані від 2 до 400 см. На показники датчика не впливає сонячне випромінювання та електромагнітні шуми.

Для підключення до схеми датчик оснащений 4 виводами. Два з них слугують для підключення живлення: Vcc та Gnd. На вхід Trig подається пусковий імпульс, а з виходу Echo знімається сигнал, довжина якого пропорційна виміряній відстані [9]. Схема підключення ультразвукового модуля до Arduino(рис. 5).

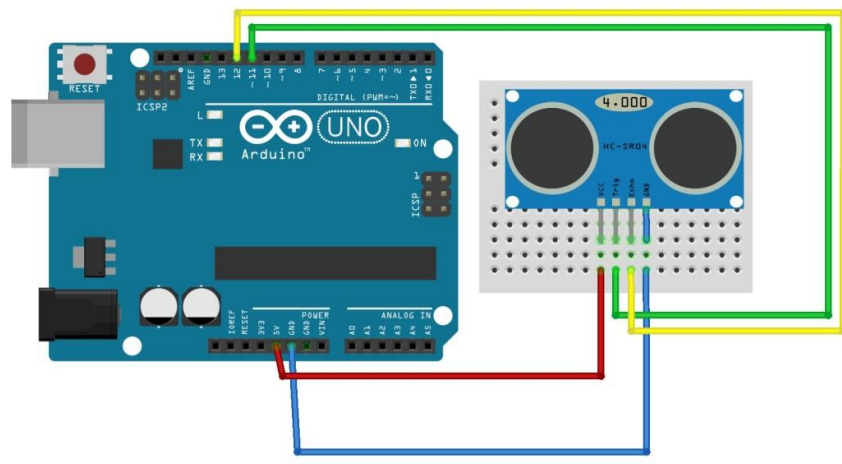


Рис. 5. Схема підключення HC-SR04 до Arduino

Фрагмент коду використання ультразвукового датчика:

```
unsigned long distance(){  
    int duration, cm;  
    digitalWrite(SONICTRIG, LOW);  
    delayMicroseconds(2);  
    digitalWrite(SONICTRIG, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(SONICTRIG, LOW);  
    duration = pulseIn(SONICECHO, HIGH);  
    return duration / 58;  
}
```

Після ввімкнення робототехнічна система визначає напрямок до джерела світла завдяки чотирьом фоторезисторам, які розміщені попереду, позаду та по боках корпусу робота. Напрямок обирається по найменшому опорі фоторезистора. Система починає рух до джерела світла. Під час руху сервопривід в невеликих межах (10°) повертає ультразвуковий датчик, який вимірює відстань до перешкоди. Якщо перешкода попереду знаходиться в заданому діапазоні відстаней, то сервопривід робить більший поворот щоб скоригувати напрям руху для оминання її. При вирівнюванні опорів фоторезисторів попарно: лівого та правого, переднього та заднього ціль вважається досягнутою. Робототехнічна система зупиняється та заряджається від джерела світла. При виході опорів фоторезисторів за межі гістерезису цикл роботи робота повторюється. Дана система може бути модернізованою під конкретні умови (рис. 6).

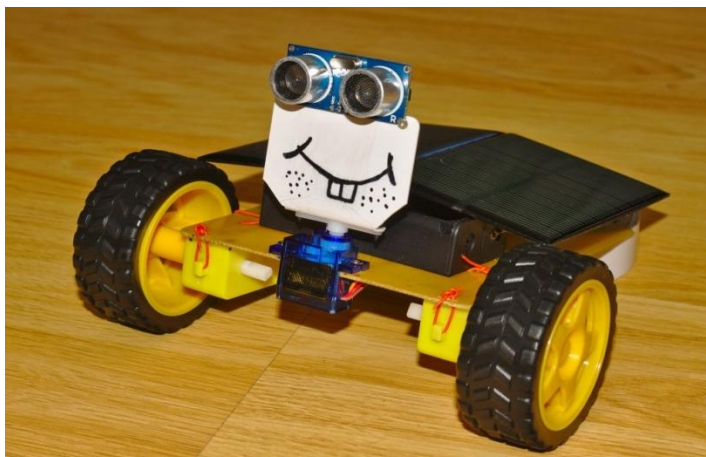


Рис. 6. Робототехнічна система автоматичного пошуку джерела світла

Висновки. Здійснено реалізацію робототехнічної системи на базі платформи Arduino Uno R3 CH340 (ATmega328). Дана система може автоматично шукати світло, перемішуватися оминаючи перешкоди та підзаряджати вбудовані акумулятори.

Список використаних джерел.

1. Новини робототехніки [Електронний ресурс]. – Режим доступу: <https://robotics.ua/news> (Дата звернення 29.03.17 р.).
2. Роботи для розвитку дітей [Електронний ресурс]. – Режим доступу: <https://babytoys.ua/uk/> (Дата звернення 30.03.17 р.).
3. Роботи — пилососи [Електронний ресурс]. – Режим доступу: <http://www.aks.ua/> (Дата звернення 25.03.17 р.).
4. BOSTON DYNAMICS [Електронний ресурс]. – Режим доступу: <https://hi-news.ru/tag/boston-dynamics> (Дата звернення 20.03.17 р.).
5. Виникнення й розвиток сучасної робототехніки [Електронний ресурс]. – Режим доступу: http://wiki.kspu.kr.ua/index.php/Виникнення_й_розвиток_сучасної_робототехніки ((Дата звернення 31.03.17 р.).
6. Автономний робот [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Автономний_робот (Дата звернення 31.03.17 р.).
7. Робототехніка в Україні: разработки и перспективы [Електронний ресурс]. – Режим доступу: <https://dou.ua/lenta/articles/robotics-in-ukraine/> (Дата звернення 31.03.17 р.).
8. Arduino для начинающих [Електронний ресурс]. – Режим доступу: <http://edurobots.ru/2014/04/arduino-servoprivod/> (Дата звернення 20.03.17 р.).
9. Ультразвуковой дальномер HC-SR04 [Електронний ресурс]. – Режим доступу: <http://amperka.ru/product/hc-sr04-ultrasonic-sensor-distance-module> (Дата звернення 21.03.17 р.).

УДК 629.113

Решетило А.О., Решетило О.М., Гуменюк П.О., Маркіна Л.М.
Луцький національний технічний університет

СИСТЕМИ ЗАБЕЗПЕЧЕННЯ МІКРОКЛІМАТУ САЛОНУ ЛЕГКОВОГО АВТОМОБІЛЯ

Решетило А.О., Решетило О.М., Гуменюк П.О., Маркіна Л.М. Системи забезпечення мікроклімату салону легкового автомобіля. У статті розглянуто основні види систем створення мікроклімату автомобіля серед яких системи кондиціонування та клімат контролю. Описано основні принципи їх роботи. Наведено ряд переваг системи клімат контролю по відношенню до систем кондиціонування.

Ключові слова: мікроклімат, системи кондиціонування, кондиціонер, клімат контроль.

Решетило А.А., Решетило А.Н., Гуменюк П.А., Маркина Л.Н. Системы обеспечения микроклимата салона легковых автомобилей. В статье рассмотрены основные виды систем создания микроклимата автомобиля среди которых: системы кондиционирования и климат контроля. Описаны основные принципы их работы. Приведен ряд преимуществ системы климат контроля по отношению к системам кондиционирования.

Ключевые слова: микроклимат, системы кондиционирования, кондиционер, климат контроль.

Reshetylo A.A., Reshetylo A. M., Gumeniuk P.A., Markina L.M. Systems of passenger microclimate cars. The article considers the main types of creating a microclimate including car air conditioning system and climate control. The basic principles of their work. Shows some advantages climate control system with respect to air conditioning systems.

Keywords: climate, air conditioning systems, air conditioning, climate control.

Однією із важливих особливостей сучасних автомобілів є підтримання мікроклімату у салоні. Адже мікроклімат виступає не лише, як елементом комфортності, а й зменшує ризик аварійності. Оскільки він впливає не лише на здоров'я, але й самопочуття, як водія, так і пасажирів. Від умов створених системами обігріву, вентиляції та кондиціонування залежать людські чинники (серед яких: втома, сонливість, апатія, ступінь уваги, швидкість зорово-рухових реакцій, здатність сприймати тимчасові інтервали, швидкість розумового процесу, асоціативна пам'ять, точність і швидкість дій та інші фактори).

Мікроклімат салону легкового автомобіля забезпечується за допомогою систем кондиціонування, які фактично складають закриту систему. Всередині неї відбувається циркуляція фреону. Власне для циркуляції фреону застосовуються компресор з приводом від колінчастого валу. Окрім цього компресор створює тиск у замкнутій системі, після чого холодоагент уже подається із компресора у газоподібному вигляді.

Проте для роботи системи кондиціонування необхідний холодоагент у рідкому стані. У зв'язку із цим він надходить до радіатора системи, де охолоджується. Після чого охолоджений фреон надходить до ресивера-осушувача (застосовується для очищення холодоагенту від домішок). Від ресивера фреон рухається до терморегуляційному вентилю, де відбувається дозування подачі фреону далі.

Кінцевим елементом системи кондиціонування є випарник. У ньому фреон знову переходить до газоподібного стану, що супроводжується сильним поглинанням тепла з повітря, яке проходить через цей радіатор. Це повітря, вже охолодженим, подається в салон автомобіля. Фреон ж у вигляді газу подається знову в компресор, і весь цикл знову повторюється.

Всі вище перераховані елементи є основними у системі кондиціонування, вони зв'язані трубопроводами, при цьому враховується, що у системі використовується різний тиск. Так, від компресора і до випарника встановлені трубопроводи високого тиску, а від випарника до компресора - низького. Крім цього система включає штуцери для управління кондиціонера, електрообладнання, тощо [10].

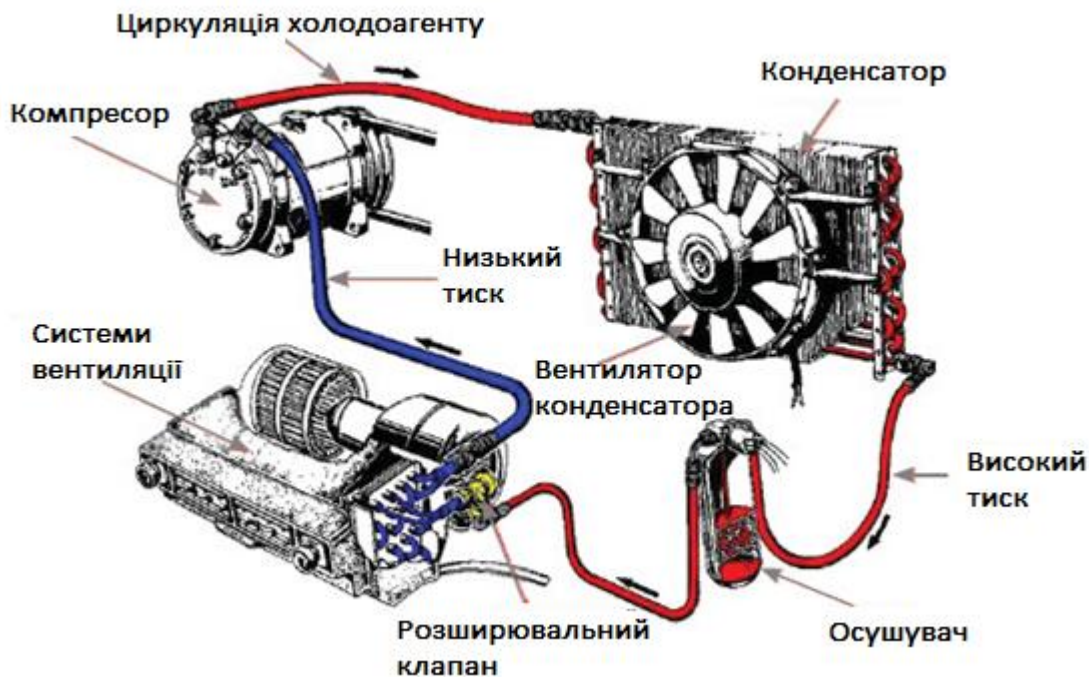


Рис. 1. Принцип роботи кондиціонера

Розглянемо більш детально на основні конструктивні елементи кліматичної системи [12].

Компресор - основний і самый складний агрегат системи. Його завдання - стиснути газоподібний холодоагент низької температури і низького тиску, перетворивши його в газ високої температури і високого тиску. В даний час в кондиціонерах застосовуються наступні типи компресорів:

- аксіально-поршневі;
- роторно-лопатеві;
- поршневі рядні і V-образні.



Рис. 2. Компресор

Вал компресора - єдине місце в системі, де розташоване "рухоме" торцеве ущільнення, що забезпечує герметичність.

Привід компресора здійснюється клиновим або поліклиновим приводним пасом від двигуна автомобіля через електромагнітну муфту. При подачі напруги на її обмотку, ведений диск і шків обертаються синхронно, приводячи в рух вал компресора.

Масло компресора створюється спеціальним компресорним маслом, що циркулює по всій системі разом з холодоагентом. У системах, що працюють з фреоном R12, застосовуються мінеральні масла, а з R134a - поліалкіленово-гліколевої (PAG). При змішуванні цих масел утворюється каламутна густа маса, яка веде до виходу з ладу системи кондиціонування, і в першу чергу компресора. При дозаправці кондиціонера холодоагентом і доливці масла використовуються тільки ті компоненти, які призначені для даної системи. Як правило, в моторному відсіку

автомобіля є наклейки, які вказують тип холодоагенту, його кількість і відповідний йому тип та кількість масла (наклейки для R134a - зеленого кольору, для R12 - жовтого).

Конденсатор - радіатор кондиціонера, як правило, алюмінієвий. У ньому відбувається конденсація (перехід в рідкий стан) нагнітається компресором холодоагенту з виділенням тепла в атмосферу. Конденсатор має додаткові електровентилятори, і для кращого обдування встановлюється перед радіатором системи охолодження. Це вразлива частина кондиціонера, що піддається не тільки механічним пошкодженням, але і корозії.



Рис. 3. Конденсатор

Випарник - теплообмінник, також, як правило, алюмінієвий. Перехід холодоагенту з рідкого стану в газоподібний (випаровування) відбувається в ньому з поглинанням тепла. Випарник розташований в салоні автомобіля, разом з радіатором обігрівача, на шляху вхідного повітряного потоку, що забезпечує зниження його температури.

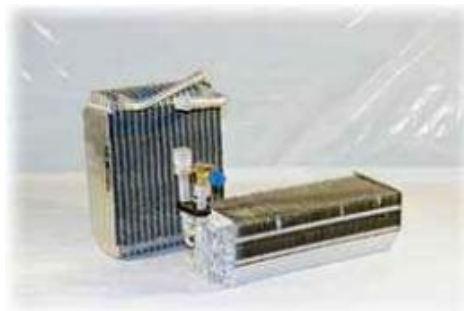


Рис. 4. Випарник

Ресивер-осушувач встановлюється на вихідному трубопроводі конденсатора перед випарником і служить резервуаром для рідкого холодоагенту, очищає його від сторонніх домішок і води. Ресивер-осушувач може забезпечуватися оглядовим вікном для контролю за кількістю холодоагенту.



Рис. 5. Ресивер-осушувач

Розширювальний клапан (РК) (може мати назву "терморегулювальний вентиль" - TRV) регулює кількість холодоагенту, що надходить у випарник. Є пристроєм, що забезпечує зміну продуктивності системи в залежності від умов і режиму роботи. Встановлюється на випарнику, рідше - в моторному відсіку на вхідному патрубку випарника.



Рис. 6. Розширювальний клапан і розширювальна трубка

Розширювальна трубка (РТ) являє собою той же розширювальний клапан, але іншої конструкції, і виконує ті ж функції, та розташована на вході у випарник. Залежно від конструкції системи кондиціонування перед випарником встановлюється РК або РТ.

Акумулятор-осушувач встановлюється на системах з розширювальною трубкою для довипарювання рідкого холодоагенту. Розташовується на трубопроводі після випарника, до компресора. Акумулятор-осушувач виконує ще й додаткові функції - осушення і фільтрації холодоагенту. При його наявності в системі ресивер-осушувач, як правило, не використовується.

Існує два види управління кондиціонером - *ручне* і *автоматичне* (клімат-контроль) [12].

Ручне управління здійснюється за допомогою: кнопки (А / С) управління компресором, ручки регулювання температури із синьою та червоною зонами, перемикачем обертів електродвигуна вентилятора кондиціонера, важелів розподілу потоків повітря по салону (вгору, вниз, центральна частина).

На роботу системи впливають наступні чинники: швидкість автомобіля, температура зовнішнього повітря, сонячна радіація, обороти двигуна і, відповідно, компресора. Недоліком ручного управління системою кондиціонування є постійне налаштування водієм комфортного клімату в салоні автомобіля. При цьому це викликає не лише дискомфорт, але й змушує водія відволіктися під час руху автомобіля.

Автоматичне управління кондиціонером виконується блоком комфорту автомобіля. При цьому водієві вистачить один раз задати на блоці управління значення бажаної температури. Система самостійно буде підтримувати і корегувати температуру, незалежно від зовнішніх і внутрішніх факторів.

Система автоматичного управління включає в себе датчики температури, електронні блоки, електричні виконавчі механізми (клапани, заслінки, вентилі). У блоці управління є також функція ручного регулювання обертів вентилятора і режимів розподілу потоків повітря у різних зонах салону автомобіля. Водій, змінюючи ці параметри на свій розсуд, виводить систему з режиму автоматичного підтримання мікроклімату.

Варто виділити також основні виконавчі механізми та датчики системи автоматичного керування кондиціонером [12]. До них належать: датчик температури зовнішнього повітря, датчик температури повітря, датчик температури внутрішнього повітря, датчик сонячної радіації, заслінка змішування, електромагнітний клапан, заслінка свіжого повітря, електромагнітна муфта компресора.

Датчик температури зовнішнього повітря розташований в передній частині автомобіля.

Датчик температури повітря, що виходить безпосередньо з кліматичної установки, розташований в корпусі випарника або в повітроводах.

Датчик температури внутрішнього повітря розташований, як правило, в центральній частині панелі приладів.

Датчик сонячної радіації знаходиться в салоні автомобіля в районі вітрового скла, над панеллю приладів.

Заслінка змішування регулює потік повітря через радіатор опалення і випарник кондиціонера. При одному крайньому положенні заслінки - режим максимального нагрівання, при іншому - режим максимального охолодження. Проміжні положення заслінки забезпечують змішування гарячого і холодного повітря в різних пропорціях.

Електромагнітний клапан встановлюється на магістралі системи обігріву салону. Призначений для зниження продуктивності кліматичної установки в режимі обігріву.

Заслінка "свіжого повітря" регулює кількість повітря, що поступає в кліматичну установку незалежно від швидкості автомобіля.

Електромагнітна муфта компресора забезпечує включення і відключення компресора кондиціонера блоком автоматичного управління.

На сьогоднішній день обов'язковою системою практично кожного автомобіля не залежно від класу є наявність у ньому системи підтримання клімату. Однак вони відрізняються принципом роботи, комплектації та функціональними можливостями.

Сучасні кліматичні установки складаються в значній мірі із електронного блоку управління клімат контролем, який на основі показів датчиків розміщених в салоні автомобіля контролює роботу кондиціонера, обігрівача та фільтруючих елементів. При цьому створюється комфортний для водія та пасажирів «клімат».

Система в змозі також визначати такі параметри, як температура зовнішнього повітря та рівень сонячної радіації. У залежності від цих показників залежить напрямки потоку повітря в різних зонах системи контролю. При цьому забезпечується рівномірний розподіл потоку і холоду по усьому середовищі. До переваг цієї системи можна віднести наступне: подача нагрітого та охолодженого повітря відбувається після змішування їх в повітропроводах і вже після цього розподіляється по салоні. Регулювання положення повітряних заслінок може відбуватися автоматично від блоку управління клімат контролем за сигналами датчиків, або вручну. Це призведе до зміни гарячого, холодного і змішаного повітряного потоку під час проходження через заслінки.

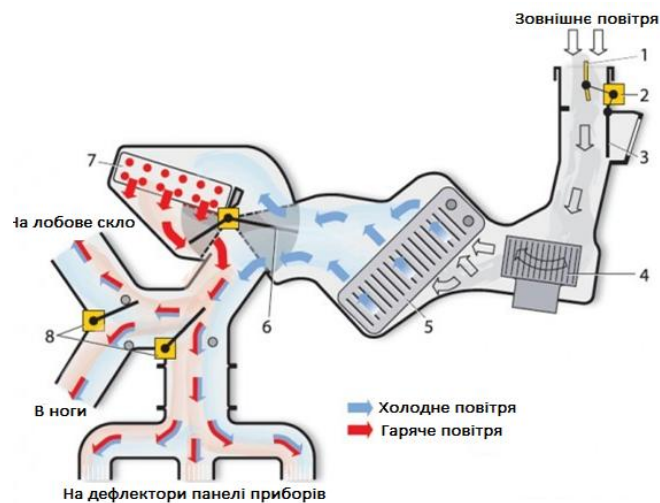


Рис. 7. Змішування потоків холодного і гарячого повітря перед подачею в салон:

1- заслінка подачі свіжого повітря; 2- привід заслінок; 3- заслінка рециркуляції; 4- вентилятор; 5- випарник; 6- температурна заслінка; 7- радіатор нагрівача; 8- заслінки зон обдуву.

Розглядаючи системи кондиціонування варто відмітити дві найбільш поширені [11]: кондиціонер і клімат контроль. Кожна із них виконує однакову функцію. Однак кліматична система має ряд наступних переваг:

- автоматично регулює і підтримує встановлену водієм температуру в салоні;
- безпечна для здоров'я пасажирів, оскільки забезпечує селективну спрямованість повітряного потоку;
- виконує регулювання не тільки температури, але і вологості, запобігаючи запотівання скла і усуваючи задуху в салоні;

- можливість включення додаткових функцій, наприклад, включення рециркуляції повітря без забору ззовні, коли воно задимлене, або ввімкнення освіжувача, ароматизованого повітря.

Варто відмітити також і особливості кліматичної системи, яка підтримує мікроклімат у салоні автомобіля не залежно від температури зовнішнього середовища на сталому рівні:

- швидке осушення повітря в кабіні в дощову і холодну погоду, щоб запобігти запотівання скла;
- очищення повітря салону від пилу і диму в спекотному і загазованому місті;
- можливість тілу людини поступово адаптуватися до прохолоди при посадці в салон авто;
- можливість працювати без забору повітря зовні.

Найпростіша схема кліматичного контролю побудована за принципом забезпечення однакової температури у всьому просторі салону [11]. Така схема називається однозонною системою забезпечення мікроклімату. Існують більш складні схеми 2-х, 3-х і 4-х зонні системи контролю клімату, в яких контроль забезпечується по окремих зонах.

Багатозонний контроль клімату являє собою систему забезпечення і підтримання різної температури в так названих зонах - посадкових місцях. Найбільш поширеною та найдешевшою є система двох зонного клімат контролю. У цій схемі є можливість встановлення різної температури в зоні крісла водія і в зоні переднього пасажирів. Особливістю такої системи є наявність обмеження на максимальну різницю температур між сусідніми зонами контролю. Вона зазвичай невелика і не перевищує 6 градусів. Це означає, що якщо водій встановив для своєї зони 30 °C, то пасажир не зможе виставити у себе менше ніж 24 °C.

Аналогічно двохзонній клімат-системи працюють 3-х та 4-х зонні системи контролю клімату. Відмінністю у них є лише наявність більшої кількості процедур, що виконуються блоком управління та забезпечення комфорту в салоні автомобіля.

Висновок. Для підтримання мікроклімату в салоні автомобіля використовуються ... До переваг відносяться.... Вони мають ряд недоліків, зокрема

Однак, при різниці температур зовнішнього повітря та повітря в салоні автомобіля більше 7-10 °C досить часто виникають алергічні реакції та запалення легень у водія та пасажирів автомобіля, що не враховує жодна з вище розглянутих систем забезпечення мікроклімату в салоні автомобіля. Тому необхідно розробити систему, яка б мала переваги над усіма системами, усувала всі недоліки та враховувала вище описане.

1. <https://vozduhoved.com/blog/mikroklimat-v-avtomobile>
2. EPJ Web of Conferences 25, 01077 (2012) DOI: 10.1051/epjconf/20122501077 © Owned by the authors, published by EDP Sciences, 2012
3. Учебное пособие «Легковые автомобили Климатические установки Вводная документация» Выпуск 04/99 ЗАО Мерседес-Бенц Автомобили Учебный центр
4. Программа самообучения 301 Touareg Климатическая установка Устройство и принцип действия
5. Automotive Climate Control Tsutomu Tabe, Katsumasa Matsui, Tohru Kakehi, and Masahiro Ohba
6. «Battery Lifetime-Aware Automotive Climate Control for Electric Vehicles» Korosh Vatanparvar, Mohammad Abdullah Al Faruque Department of Electrical Engineering and Computer Science University of California, Irvine, Irvine, California, USA
7. <http://ecouniver.com/8610-mikroklimat-v-kabinax-avtomobilya.html>
8. <http://carmp3.com.ua/vozduh-v-salone/mikroklimat-v-avtomobile.html>
9. http://im.lipetsk.ru/a_mo_kondicioner.htm
10. <http://avtomotoprof.ru/obsluzhivanie-i-uhod-za-avtomobilem/remont-avtokonditsionerov-prichinyi-vozniknoveniya-neispravnostey-i-sposobyi-borbyi-s-nimi/>
11. <http://avtomotoprof.ru/elektronnyie-sistemyi-avtomobilya/klimat-kontrol-ili-konditsioner/>
12. http://www.termotransport.com/cond/a_cond.htm

УДК 004.5

Шабелюк О.В.

Київський національний університет імені Тараса Шевченка

РОЗРОБКА БІБЛІОТЕКИ ФІЗИЧНОГО ОБЛАДНАННЯ ДЛЯ ПОБУДОВИ НАВЧАЛЬНОЇ СИСТЕМИ НА ОСНОВІ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

Шабелюк О.В. Розробка бібліотеки фізичного обладнання для побудови навчальної системи на основі доповненої реальності. В даній роботі розглядаються існуючі підходи до реалізації систем мобільного навчання фізичним дисциплінам. Пропонується використання доповненої реальності, для реалізації інтерактивного виконання практичних робіт. Особлива увага приділяється створенню бібліотеки фізичного обладнання, що може використовуватись для візуалізації моделей реального фізичного обладнання та взаємодії між ними. Описаний процес взаємодії змодельованого обладнання на прикладі моделі лазера та моделі дифракційної ґратки. Проаналізовано переваги та недоліки такого підходу, можливості до застосування для моделювання інших фізичних експериментів.

Ключові слова: мобільне навчання, віртуальна лабораторія, доповнена реальність, дистанційна фізична освіта.

Шабелюк А.В. Разработка библиотеки физического оборудования для реализации учебной системы на основе дополненной реальности. В данной работе рассматриваются существующие подходы к реализации систем мобильного обучения физическим дисциплинам. Предлагается использование дополненной реальности для реализации интерактивного выполнения практических работ. Особенное внимание уделяется созданию библиотеки физического оборудования, что может использоваться для визуализации моделей реального физического оборудования и взаимодействия между ними. Описанный процесс взаимодействия смоделированного оборудования на примере модели лазера и модели дифракционной решетки. Проанализированы преимущества и недостатки такого подхода, возможности для использования и моделирования других физических экспериментов.

Ключевые слова: мобильное обучение, виртуальная лаборатория, дополненная реальность, дистанционное физическое образование.

Shabeliuk O.V. Development of library of physical equipment for building educational system based on augmented reality. This paper examines existing approaches to the implementation of mobile learning systems of physical disciplines. It is proposed to use augmented reality for the implementation of practical interactive work. Particular attention is paid to the library of physical equipment, that can be used to visualize models of actual physical hardware, and the interaction between them. Described the interaction of simulated equipment, on the example of laser and diffraction grating patterns. Analyzed advantages and disadvantages of this approach, the possibilities for use in the simulation of other physical experiments.

Keywords: m-learning, virtual laboratory, augmented reality, distance physics education.

Постановка наукової проблеми. Мобільне навчання є одним з найперспективніших напрямків дослідження в галузі дистанційної освіти, оскільки воно забезпечує можливість навчатись будь-де і у будь-який час. Актуальність даної тематики полягає в двох основних аспектах:

По перше: стрімкий розвиток технологій виробництва за останні роки забезпечує зниження ціни та покращення технічних характеристик смартфонів та планшетних комп'ютерів – основних пристроїв забезпечення процесу мобільного навчання, і, як наслідок, їхня кількість серед населення планети продовжує стрімко рости.

По друге: спостерігається розвиток інфраструктури покриття та швидкості доступу до мережі інтернет, що забезпечує можливість регулярного швидкого обміну даними між сервером системи мобільного навчання та клієнтами – мобільними пристроями.

Методологія класичного навчання студентів технічних спеціальностей побудована за принципом вивчення лекційного матеріалу, розв'язування прикладних задач на семінарських заняттях, а також практичного закріплення вивченого матеріалу під час лабораторних курсів.

На сьогоднішній день створено багато систем мобільного навчання фізичним дисциплінам, деякі із них описані в [1-7]. Вони побудовані на принципі надання доступу до текстової, графічної, анімаційної, аудіо та відео інформації для самостійного опрацювання, але цього вже не достатньо. Разом з покращенням технічних характеристик мобільних пристроїв, з'являються можливості для розробки сучасних, більш ресурсоємних систем навчання, зокрема для виконання інтерактивних практичних (лабораторних) робіт.

Найголовнішою проблемою залишається реалізація інтерактивного процесу виконання практичних робіт, які згідно із досліджень автора [8] є основою розуміння сутності інформації і її збереження в довгостроковій пам'яті.

Аналіз досліджень. На сьогоднішній день існують різні підходи до розв'язання проблеми інтерактивності навчального матеріалу, зокрема, в роботах [9-15], пропонується дистанційне

виконання лабораторних робіт на реальному обладнанні через веб інтерфейс. Перевагою такого методу є отримання реальних, а не змодельованих фізичних даних експерименту, але суттєвим недоліком таких систем є відсутність можливостей постійного доступу до лабораторного практикуму та одночасного виконання робіт декількома різними студентами. Зокрема вихід з ладу системи з будь-яких причин – унеможливорює подальше виконання цієї роботи всіма студентами. Більшість систем такого типу працюють через веб інтерфейс, що дає можливість роботи з ними через мобільний пристрій, але така робота буде не зручна через відносно невеликий екран пристрою та високу деталізацію керуючого інтерфейсу системи.

В роботах [16,17] пропонується виконання віртуальних практичних робіт змодельованих на сервері, без використання реального обладнання. Такі системи потребують постійного інтернет зв'язку з сервером та не є достатньо інтерактивні, для відчуття роботи з реальними технічними пристроями.

В роботах [18,19,20] пропонується робота з навчальними системами, що дозволяють інтерактивно моделювати різні фізичні системи, але для цього використовуються різне додаткове обладнання, що йде в супереч концепції мобільного навчання.

В роботі [21] автори пропонують моделювати фізичні процеси у вигляді інтерактивної гри. Такий підхід до навчання є ефективним для закріплення і розуміння теоретичного курсу, але не дає змоги отримати практичні навички роботи з фізичним обладнанням.

Після аналізу наведених вище робіт, можна виділити основні особливості, які повинна мати інтерактивна система мобільного навчання для студентів технічних спеціальностей:

- Має моделювати реальне фізичне обладнання, працювати за реальними законами фізики;
- Має бути кроссплатформенна, для роботи з більшістю типів мобільних пристроїв;
- Процес виконання роботи та керування обладнанням має бути простим та зрозумілим для користувача.

Виклад основного матеріалу й обґрунтування отриманих результатів. Щоб побудувати інтерактивну систему, що задовольняє описані вимоги пропонується використання технології доповненої реальності.

Доповнена реальність – термін, який описує процес доповнення реальності будь-якими віртуальними об'єктами. При цьому робота з об'єктами може виконуватись в режимі реального часу, а для забезпечення цього процесу необхідна лише веб камера – саме зображення з якої буде доповнюватись віртуальними об'єктами (див. рис. 1 та рис. 2).

За словами дослідника Рональда Азуми [22] доповнену реальність можна визначити як систему, яка:

1. Поєднує віртуальне та реальне;
2. Взаємодіє в реальному часі;
3. Працює в 3D.



Рис. 1. Концепція доповненої реальності

На сьогоднішній день можна з легкістю назвати декілька поширених прикладів використання технології доповненої реальності. Зокрема телевізійний прогноз погоди, в якому відео потік ведучого накладається на комп'ютерне зображення мапи з погодою. Рахунок матчу, відстань від точки виконання штрафного удару чи лінія офсайду в футбольних телетрансляціях.

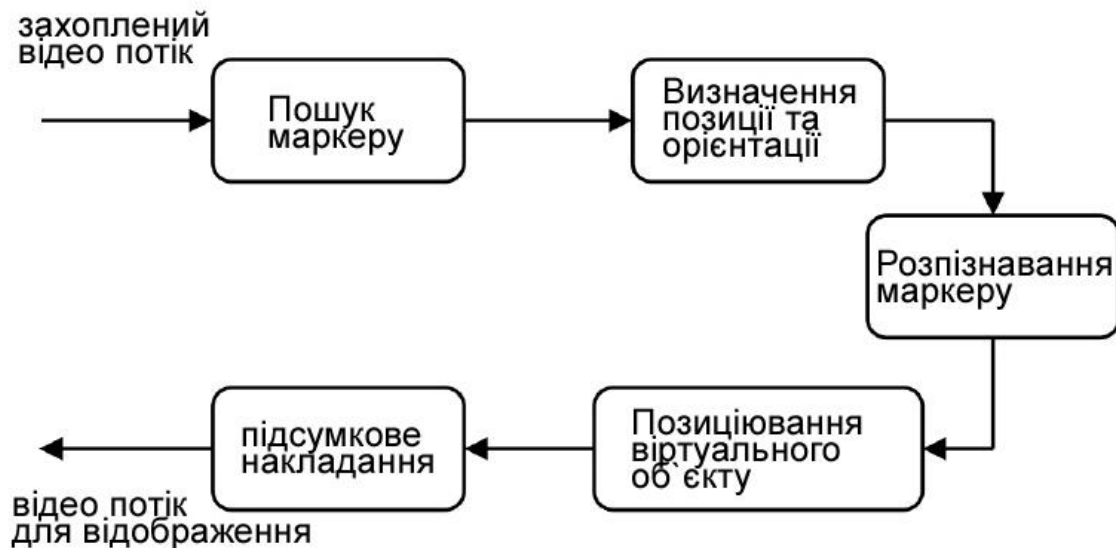


Рис. 2. Схема доповненої реальності

Використання доповненої реальності для створення інтерактивної мобільної навчальної системи дозволить візуалізовувати та взаємодіяти із фізичним обладнанням максимально наближено до реального процесу виконання фізичного експерименту.

Для реалізації технології доповненої реальності в програмному прототипі мобільної системи дистанційного навчання фізичним дисциплінам було використано платформу Vuforia - платформа доповненої реальності яка використовує технології комп'ютерного зору для відслідковування плоских зображень (маркерів). Данна платформа надає можливість розробнику інтегрувати доповнену реальність в мобільний програмний продукт, на платформах Android, IOS та Windows Phone, а також працювати із крос платформним інструментом Unity. Unity це програмний продукт, що використовується для розробки два та тривимірних застосунків реалізуючи сценарії на C#, JavaScript та Boo.

Для побудови програмного прототипу мобільної початкової системи було використано саме Unity, що зумовлено наступними факторами:

- кросплатформність Unity, що дозволяє розробити єдиний мобільний додаток, який зможе легко перебудовуватись та працювати під основними мобільними платформами;
- можливість використовувати імпортовані 3D зображення від різноманітних редакторів;
- легкість роботи із 3D зображеннями, їх розмірами та координатами.

Бібліотека моделей фізичного обладнання буде складатись із наборів моделей (3D об'єктів фізичного обладнання), відповідних їм маркерів попередньо заданих розмірів (10x10 сантиметрів) та скрипта, що описує фізичну взаємодію цих віртуальних моделей (див. рис. 3). Для того, щоб додати новий об'єкт в бібліотеку, необхідно створити для нього маркерне зображення, 3D модель цього об'єкту, та описати його фізичні властивості та алгоритми взаємодії із іншими моделями в бібліотеці.

Після появи в відео потоці маркеру, відповідна йому 3D модель фізичного обладнання буде масштабована та зорієнтована відповідно до положення та розмірів маркеру, а потім одразу накладена на відео потік (див. рис. 4). Якщо у відео потоці опиняться 3 різних маркери, програма зможе працювати із ними одночасно, а при зміні положення будь-якого із маркерів – аналогічно змінюється положення відповідної йому 3D моделі.

Скрипт на C# - є головним програмним кодом взаємодії 3D об'єктів та реалізації фізичних законів, що ми моделюємо. В ньому описані всі властивості всіх об'єктів, доданих до бібліотеки фізичного обладнання.

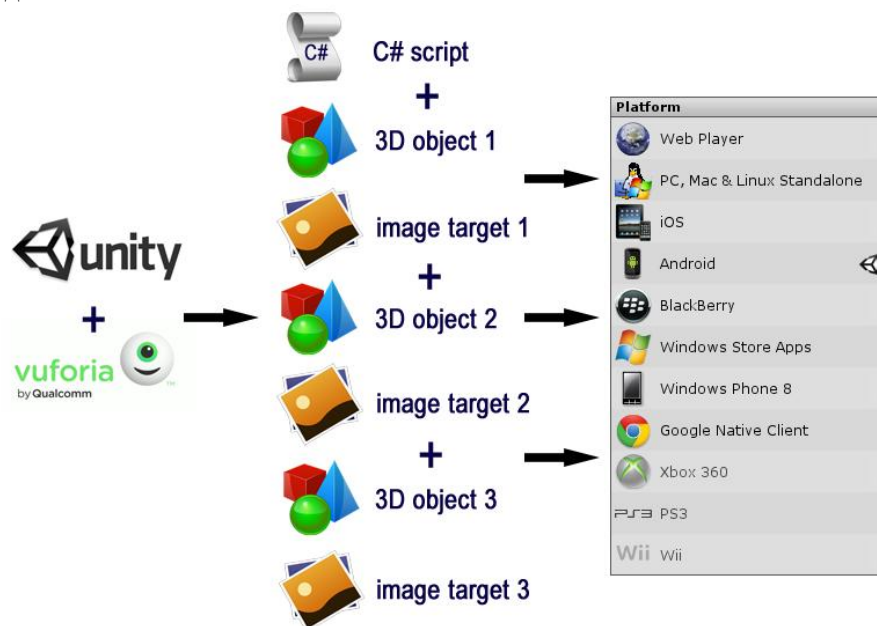


Рис. 3. Схема розробки навчальної системи: інструменти розробки, елементи бібліотеки фізичного обладнання, можливі платформи для фінальної компіляції (з ліва на право)

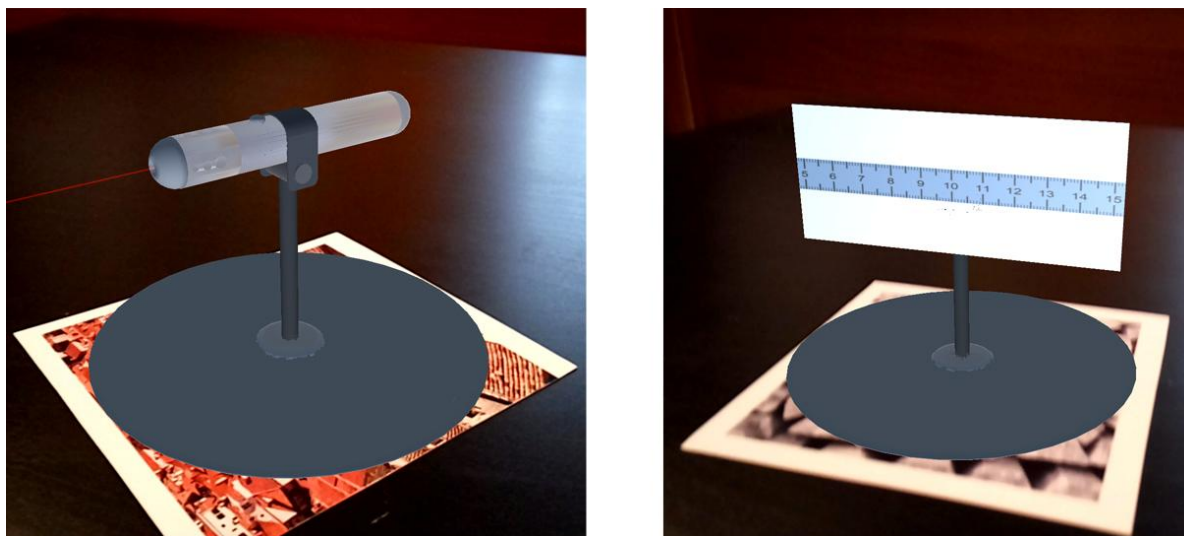


Рис. 4. Масштабовані та зорієнтовані по маркеру 3D моделі (лазер з ліва, екран з права)

Наприклад модель лазера має наступні властивості:

- Включати/виключати лазерне випромінювання в залежності від натискання на кнопку живлення лазера, що розташована зверху;
- Лазерне випромінювання – розповсюджується прямолінійно, до зіткнення із будь-яким об'єктом, при зіткненні перевіряється з яким саме об'єктом відбулось зіткнення променя, і якщо його немає в бібліотеці, то подальше розповсюдження променя припиняється. Якщо об'єкт знайдений в бібліотеці, то подальше розповсюдження променя буде відбуватись по описаному в бібліотеці фізичному закону, тобто якщо це наприклад скло – то промінь буде заломлюватись, якщо це дифракційна ґратка – то буде спостерігатись декілька дифракційних максимумів, а якщо це дзеркало – то відбиватись

і так далі. Якщо на шляху променю не потрапить жоден об'єкт – то він просто буде світити прямолінійно на нескінченність;

- Положення джерела випромінювання відносно стійки лазера, що дозволяє точно вимірювати відстань від джерела випромінювання до об'єкту зіткнення.
- Також лазерне випромінювання має свої характеристики, такі як наприклад довжина хвилі випромінювання, від якої залежить якого типу лазер використовується і відповідно змінюється колір випромінювання. Також ця характеристика буде враховуватись при розрахунках фізичних процесів, таких як дифракція, оскільки безпосередньо на них впливає. Такі параметри можна змінювати під час роботи для спостереження реакції системи на їх зміну або більш точного теоретичного розрахунку інших невідомих параметрів.

На рис. 5 зображено знімок з екрану мобільного телефону при тестуванні прототипу системи. На ньому можемо бачити як лазерний промінь, що проходить крізь дифракційну ґратку, утворює систему максимумів, умовно зображених променями лазера після проходження ґратки. Сам екран не містить жодних алгоритмів взаємодії із лазерним випромінюванням, тому просто не дає йому розповсюджуватись далі, на відміну від тої частини, котра не потрапила на екран.

Дифракційна ґратка має свої параметри, такі як: період, роздільна здатність, кутова дисперсія, тощо. Всі ці параметри впливають на подальше розповсюдження лазерного променю і спостереження дифракційних максимумів на екрані.

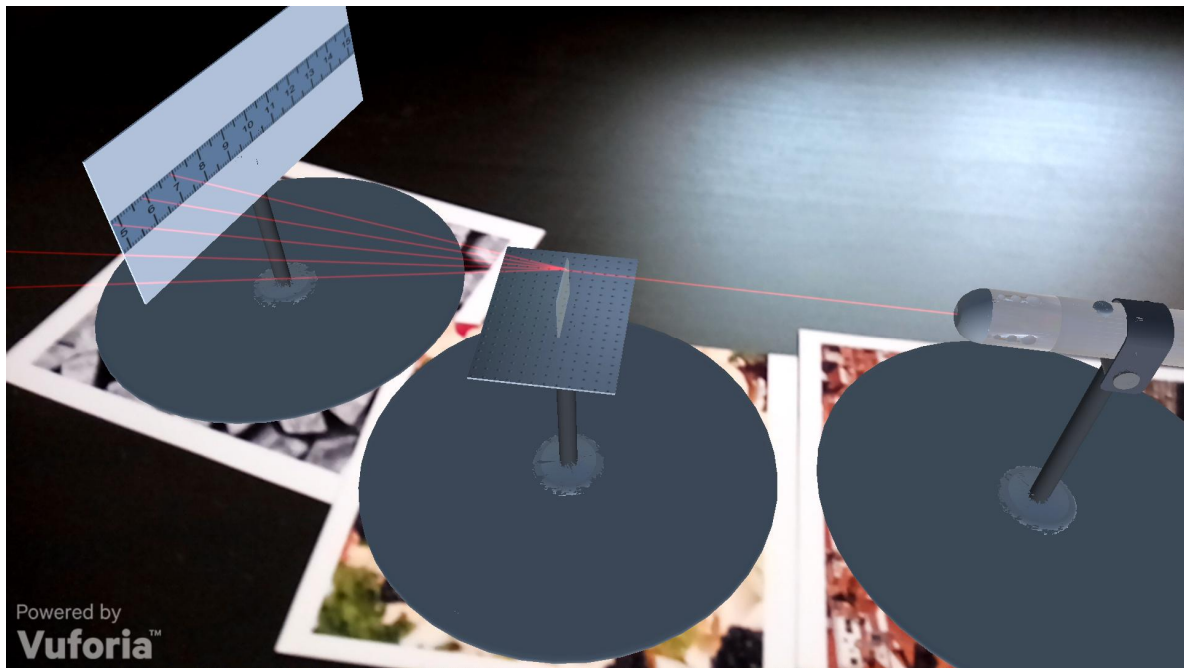


Рис. 5. Хід лазерного променю регулюється скриптом фізики

Користувач може змінювати відстань між дифракційною ґраткою та екраном вручну – змінюючи відстань між їхніми маркерами, що призведе до збільшення відстані між відповідними 3D моделями і, як наслідок, зміниться положення дифракційних максимумів, що спостерігаються на екрані.

Описавши алгоритми фізичної взаємодії 3D моделей реального фізичного обладнання і об'єднавши їх в бібліотеку - можна використовувати її для створення інтерактивних мобільних навчальних систем із реалізацією практичних робіт.

Висновки та перспективи подальшого дослідження. Актуальність впровадження технології доповненої реальності в освітній процес полягає в тому, що використання такої новітньої системи безсумнівно збільшить мотивацію студентів, а також підвищить рівень засвоєння інформації за рахунок різноманітності і інтерактивності її візуального представлення.

Використання таких систем дозволить перенести частину практичних робіт в дистанційну форму виконання та надасть змогу пройти курс всім охочим, як додатковий, навіть якщо він не передбачений навчальною програмою.

Розробка бібліотеки фізичного обладнання дозволяє повторно використовувати вже описані фізичні моделі, при розробці нових практичних курсів, проводити моделювання вільних фізичних експериментів із задіянням будь-якої кількості елементів будь-якого типу обладнання із описаного в бібліотеці набору.

Можливість змінювати деякі параметри фізичних об'єктів із бібліотеки – персоналізує виконання роботи та дозволяє реалізувати повторне виконання роботи без втрати цікавості до процесу її виконання.

Кросплатформність такої системи дозволить використовувати її на будь-якому мобільному пристрої що має камеру, для цього необхідно лише роздрукувати маркери приладів.

Недоліком такої системи є складність додавати нові об'єкти фізичного обладнання із їхніми фізичними властивостями, адже необхідно описати всі властивості нового об'єкта та алгоритм його взаємодії із існуючими об'єктами в бібліотеці, якщо така взаємодія теоретично можлива.

1. J. Borondo, R. M. Benito, and J. C. Losada, "Adapting physics courses in an engineering school to the b-learning philosophy," *European Journal of Engineering Education*, vol. 39, no. 5, pp. 496–506, 2014.
2. T. Mart'in-Blas and A. Serrano-Fern'andez, "The role of new technologies in the learning process: Moodle as a teaching tool in physics," *Computers & Education*, vol. 52, no. 1, pp. 35–44, 2009.
3. R. V'asquez-Ram'irez, G. Alor-Hern'andez, and A. Rodr'iguez-Gonz'alez, "Athena: A hybrid management system for multi-device educational content," *Computer Applications in Engineering Education*, vol. 22, no. 4, pp. 750–763, 2012.
4. K. Lassila-Perini, S. Malik, B. Hegner, A. Hinzmann, and R. Wolf, "Planning and organization of an e-learning training program on the analysis software in CMS," in *Proceedings of the "International Conference on Computing in High Energy and Nuclear Physics (CHEP 2010)"*, IOP Publishing, 2010.
5. K. Verma, S. Dubey, and M. A. Rizvi, "Mobile cloud a new vehicle for learning: m-learning its issues and challenges," *International Journal of Science and Applied Information Technology*, vol. 1, no. 3, pp. 113–131, 2012.
6. T. Hasegawa, H. Kojima, C. Masu, Y. Fukushima, H. Kojima, K. Konokawa, T. Isobe, E. Sato, H. Murayama, K. Maruyama, and T. Umeda, "Creation and application of three-dimensional computer-graphic animations for introduction to radiological physics and technology," *Radiol Phys Technol*, vol. 3, no. 1, pp. 1–9, 2010.
7. J. Keengwe and M. Bhargava, "Mobile learning and integration of mobile technologies in education," *Education and Information Technologies*, vol. 19, no. 4, pp. 737–746, 2014.
8. B. Faghhi, M. R. Azadehfar, and S. D. Katebi, "User interface design for elearning software," *The International Journal of Soft Computing and Software Engineerin*, vol. 3, no. 3, pp. 786–794, 2013.
9. J. V. Nickerson, J. E. Corter, S. K. Esche, and C. Chassapis, "A model for evaluating the e-effectiveness of remote engineering laboratories and simulations in education," *Computers & Education*, vol. 49, no. 3, pp. 708–725, 2007.
10. X. Chen, G. Song, and Y. Zhang, "Virtual and remote laboratory development: A review," in *Earth and Space 2010: Engineering, Science, Construction, and Operations in Challenging Environments*, pp. 3843–3852, American Society of Civil Engineers, 2010.
11. T. Eppes, P. Schuyler, and T. Oruganti, "Pilot test results of a new distance laboratory platform," in *Proceedings of the 2005 Annual Conference & Exposition, Session 2550*, 2005.
12. A. Bargelis, R. Mankute, and M. Rima'sauskas, "Virtual and distance labs for vocational education training of industrial employees," in *Proceedings of the "18th International Conference. Mechanika 2013"*, pp. 21–26, Kaunas University of Technology, 2013.
13. Z. Nedic1, J. Machotka, and A. Nafalski, "Remote laboratories versus virtual and real laboratories," in *Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference T3E-3*, pp. 1–6, IEEE, 2003.
14. E. Scanlon, C. Colwell, M. Cooper, and T. D. Paolo, "Remote experiments, reversioning and re-thinking science learning," *Computers & Education*, vol. 43, no. 1-2, pp. 153–163, 2004.
15. S. Khachadorian, H. Scheel, P. de Vries, and C. Thomsen, "A practical approach for applying online remote experiments: Onprex," *European Journal of Engineering Education*, vol. 36, no. 1, pp. 21–34, 2014.
16. E. Costello, S. Fox, and T. Lynn, "Online labs for distance learners: reflections from an irish pilot study," in *Proceedings of the "International Symposium for Engineering Education WISEE-08"*, Dublin City University, 2008.
17. S. J. Elliott and E. P. Kukula, "The challenges associated with laboratory-based distance education," *EDUCAUSE Quarterly*, no. 1, pp. 37–42, 2007.
18. X. PingJun, A. M. Lopes, M. T. Restivo, and Y. YingXue, "A new type haptics-based virtual environment system for assembly training of complex products," *The International Journal of Advanced Manufacturing Technology*, vol. 58, no. 1-4, pp. 379–396, 2011.
19. J. T. Jones and M. Joordens, "Distance learning for laboratory practical work in microcontrollers," *Distance Learning for Laboratory Practical Work in Microcontrollers*, vol. 19, no. 3, pp. 455–459, 2003.
20. H. Kaufmann and B. Meyer, "Simulating educational physical experiments in augmented reality," in *Proceedings of the "ACM SIGGRAPH ASIA 2008 educators program"*, ACM Press, 2008.
21. J. L. Anderson and M. Barnett, "Learning physics with digital game simulations in middle school science," *Journal of Science Education and Technology*, vol. 22, no. 6, pp. 914–926, 2013.
22. R. T. Azuma, "A survey of augmented reality," in *Presence: Teleoperators and Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.