

DOI: <https://doi.org/10.36910/6775-2524-0560-2022-49-09>

УДК 004.41

Поплавська Ганна Володимирівна¹, викладач

<https://orcid.org/0000-0002-1185-8868>

Сачук Юрій Володимирович², к.т.н., доцент

<https://orcid.org/0000-0002-1317-1103>

¹ Комунальний заклад вищої освіти «Луцький педагогічний коледж» Волинської обласної ради, м. Луцьк, Україна

² Луцький національний технічний університет, м. Луцьк, Україна

МЕТОДИЧНІ АСПЕКТИ НАВЧАННЯ СТУДЕНТІВ РОЗРОБКИ ДОДАТКУ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ УЧНІВ ІЗ ВИКОРИСТАННЯМ LASARUS

Поплавська Г. В., Сачук Ю.В. **Методичні аспекти навчання студентів розробки додатку для перевірки знань учнів із використанням Lazarus.** У статті розкрито особливості створення додатку для перевірки знань учнів із використанням мови програмування Lazarus. На прикладі створення тестового додатку, розробленого у середовищі програмування Lazarus, розглянуто основні етапи проектування, об'єкти та алгоритмічні конструкції середовища.

Ключові слова: Lazarus, тестування, проєкт, тест, перевірка знань, програмування.

Poplavska G., Sachuk Yu. Methodical aspects of teaching students to develop an application for testing students' knowledge using Lazarus. The article describes the features of creating an application for testing students' knowledge using the Lazarus programming language. On the example of creating a test application developed in the Lazarus programming environment, the main stages of design, objects and algorithmic designs of the environment are considered.

Keywords: Lazarus, testing, project, test, knowledge test, programming.

Постановка проблеми. Перевірка знань, вмінь та навичок учнів у засвоєнні тем з певних предметів, особливо початкової школи, стала вагомим складовим навчання. Результат навчання великою мірою залежить від методів контролю знань. Адже, контроль знань – це засіб управління навчальною діяльністю учнів. Для того, щоб з функцією перевірки реалізувалася і функція навчання, необхідно створити певні умови та використовувати певні засоби об'єктивної перевірки знань. Для підвищення ефективності навчального процесу, засоби перевірки знань, умінь і навичок учнів початкової школи мають зацікавити здобувачів освіти своїм інтерфейсом, правильно та коректно сформульованими запитаннями.

Аналіз досліджень. Питання застосування тестування учнів вивчається уже багато років. Вагомим дослідженням в цій галузі внесли вчені В. В. Божкова, Мединська С. І., Савчин М. В. Питанням дослідження використання ІКТ- технологій тестування серед учнів займалися Н. Дичківська, О. Сікорський, Н. Щетина, Ляшенко О. І., Лукіна Т. О. [9]

Формулювання цілей статті. Мета статті – розкрити особливості розробки додатку для перевірки знань учнів із використанням Lazarus.

Виклад основного матеріалу. Важливою складовою ланкою навчання є перевірка знань, вмінь та навичок учнів, на основі якої можна виявити переваги та недоліки знань учнів, що в свою чергу призводить до оптимального підбору форм та методів для подання матеріалу вчителем-предметником. Сьогоднішнє стрімке реформування в освіті посприяло використанню ширшим колом вчителів засобів тестування, як ефективного засобу визначення рівня навчальних досягнень учнів. Цей засіб уже багато років зарекомендував себе, як ефективний в теоретичному та практичному плані. На сучасному етапі розвитку освіти тестування набуває визначального характеру в процесі з'ясування компетентності учнів.

Дискусії фахівців з приводу доцільності чи недоцільності введення тестування у навчальні програми, щоб виявити якість освіти, ведуться й досі. Однак, зовнішнє незалежне оцінювання у тестовій формі - це наше сьогоднішнє і, найвірогідніше, майбутнє, тому варто робити «ставку» саме на цей засіб вимірювання рівня компетентності здобувача освіти.

Приступаючи до розкриття мети статті, доцільно спочатку коротко охарактеризувати Lazarus як мову програмування. Lazarus – мова візуального програмування, це візуальний компілятор, який працює у середовищі Windows на базі мови Object Pascal.

Lazarus відноситься до систем візуального програмування, які називають також системами швидкої розробки додатків [3]. Побудова додатків в Lazarus передбачає два взаємопов'язані кроки:

- побудова інтерфейсу програми;

- програмне проектування.

Стадії й етапи розробки тестової програми в середовищі Lazarus:

Перший етап - Аналіз задачі, визначення засобів та методів її вирішення .

Другий етап - Проектування загального алгоритму роботи програми.

Третій етап - Розробка функціональних алгоритмів роботи програми, побудова інтерфейсу .

Четвертий етап - Робробка програмного забезпечення .

П'ятий етап - Тестування програми .

Шостий етап - Завершення та збереження проєкту.

Побудований інтерфейс користувача встановлює взаємодію між користувачем та додатком. Зовнішній вигляд проєкту будується методом розміщення на формі компонентів, які називають об'єктами. Щоб відтворити дію роботи запланованого додатку, що розробляється, використовують вікно модуля(коду). При побудові інтерфейсу програми використовують вікно форми, вікно інспектора об'єктів та головне вікно (рис. 1).

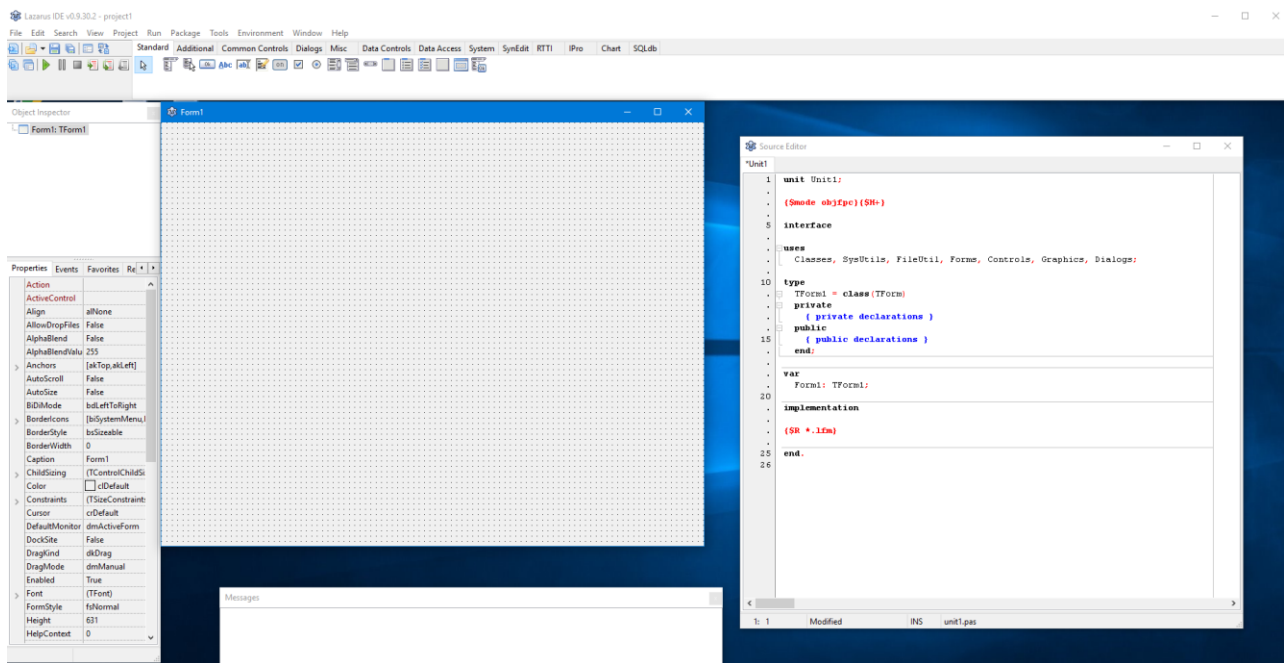


Рис. 1. Інтерфейс візуального середовища Lazarus

Вікна форми та коду тісно пов'язані між собою. Коли ми робимо якісь зміни із формою (базовим об'єктом будь-якого проєкту), то це призводить до автоматичної зміни коду програми. Тому спочатку треба сконструювати форму за тими запитами, які стоять перед розробником, розміщуючи на ній усі компоненти додатку, а потім переходити до написання коду програми, в яких поетапно відображати дії компонентів.

Коли конструюємо проєкти, то розміщуємо на формі компоненти, які підходять для відображення даних проєкту. Усі компоненти у Lazarus є об'єктами. Кожен об'єкт може виконувати певну дію, реагуючи на певну подію. Подіями є будь-які дії користувача та операційної системи щодо кожного компонента окремо.

У Lazarus компоненти – це об'єкти, що реагують на події. Компонент, розміщений на формі, як і будь-який інший об'єкт, відноситься до певного класу. Усі компоненти мають властивість Name – назву компонента. Lazarus дає кожному компоненту загальне ім'я Назва_компонента_N, де N змінюється, починаючи з 1. Найкраще змінювати ім'я, щоб воно відображало призначення даного компонента у програмі. Усі візуальні об'єкти мають різні властивості, що розташовані у Інспекторі об'єктів. Властивості можуть бути як прості, так і складені. Значення кожної властивості можна обирати із списку, що пропонується, або задавати власноруч змінюючи у вікні Інспектора об'єктів. Також можна надавати значення властивостей програмно в програмно в редакторі коду. [5]

Форма є полотном для побудови додатків, нагадує звичайне вікно Windows. На ній розташовані стандартні елементи вікна: кнопки меню, розкриття, згортання, закриття та рядок

заголовка. На формі за допомогою миші розташовуються компоненти. Форма є об'єктом класу TForm.

Форми, що відображають різноманітні повідомлення та запитують від користувача введення будь-якої інформації, називають діалоговими вікнами. Деякі форми дають можливість здійснити двосторонню передачу інформації, але головним завданням діалогових вікон є передача інформації від користувача до програми. Діалоги можна відкрити тільки в модальному режимі та на час передачі інформації.

Коли показана модальна форма, всі повідомлення, що надходять обробляються модальною формою. Вибір способу створення діалогу залежить від завдання, що розв'язується.

Головні компоненти Lazarus, які слід використати при розробці тестової програми, знаходяться на вкладках Standard та CommonControls знаходяться стандартні та додаткові компоненти палітри, а саме: Main Menu – головне меню; Button – стандартна кнопка; Label – мітка; Edit – однорядковий редактор; CheckBox -прапорець; RadioButton – залежний перемикач; RadioGroup – група залежних перемикачів; PageControl – панель.

При розробці програм часто виникає завдання об'єднати або згрупувати різні елементи управління. Об'єднання елементів виконується за допомогою компонентів –контейнерів. Контейнер є візуальним компонентом, на поверхні якого можна розміщувати інші компоненти. Форма програми є основним контейнером, з якого зазвичай і починається проектування інтерфейсу. Форма є власник всіх розміщених на ній компонентів.

При створенні графічних зображень використовуються графічний компонент Image - графічне зображення.

Компонент Image використовується для відображення зображення.

Основна властивість – Picture, що визначає зображення, яке озміщується всередині компонента зображення. Зображення з графічного файлу можна завантажувати на етапі проектування програми та при його виконанні. Потрібно враховувати, що зображення, підключене під час проектування, значно збільшує обсяг файлу програми, що виконується. Щоб уникнути цього, рекомендується завантажувати великі зображення динамічно.

Розглянемо основні можливості Lazarus як засобу розробки контролюючої програми на прикладі лабораторного практикуму з «Основ програмування та технології створення програмних продуктів».

ПРАКТИЧНЕ ЗАНЯТТЯ

Модуль VI. Програмування у середовищі розробки Lazarus.

Тема Створення контролюючої програми

Мета: формування простих навичок створення проектів з використанням декількох форм

Завдання:

- *студенти повинні знати:* інтерфейс програми, призначення, властивості, події для форм; поняття модальної та немодальної форм;
- *студенти повинні вміти:* створювати проекти з використанням декількох форм, застосовуючи при цьому окремі шаблони форм

Хід виконання:

Завдання. Створити програму-тест для учнів початкової школи, яка містить 6 запитань:

1. Скільки пір року ви знаєте? (Для введення відповіді використайте компонент Edit)
2. Які місяці є зимовими?
 - Січень, лютий, грудень
 - Березень, квітень, травень
 - Червень, липень, серпень
 - Вересень, жовтень, листопад
3. У якому місяці приходиться Святий Миколай?
 - Грудень
 - Січень
 - Лютий
 - Березень
4. В яку пору року з'являються проліски?
 - Зима
 - Весна

- Літо
 - Осінь
5. Скільки днів у високосному році? (Для введення відповіді використайте компонент Edit)
 6. Який місяць має 29 днів?
 - Березень
 - Жовтень
 - Липень
 - Лютий

Алгоритм розробки проекту

1. Створити папку D:\Прізвище\Pract_test.
2. Змінюємо назву і колір форми і вставляємо компонент *CommonControls* → *PageControl*. Розтягуємо рамку цього компонента і з контекстного меню обираємо *Add Page* (Додати сторінку) (стільки разів, скільки запитань у тестах).
3. Змінимо назви вкладок: 1 питання, 2 питання, 3 питання...
4. Переходимо на вкладку *1 питання*. Розміщуємо на цій вкладці для написання запитання елемент *Standard* → *Label*, для введення відповіді *Standard* → *Edit*. Розміщуємо ще на закладці кнопку *Standard* → *Button1*. Задаємо заголовок кнопки *Відповісти на запитання*. Аналогічно робимо із 5 питанням.
5. Переходимо на вкладку *2 питання*. Розміщуємо на цій вкладці для написання запитання елемент *Standard* → *Label*, для варіантів відповідей *Standard* → *RadioGroup*. Справа від *RadioGroup* вставляємо зображення згідно теми запитання.
6. Аналогічно заповнюємо інші вкладки з запитаннями: кожна вкладка повинна містити компоненти *Label*, *Standard*, *RadioGroup*.
7. Створюємо головне меню *MainMenu* тесту (рис. 2).

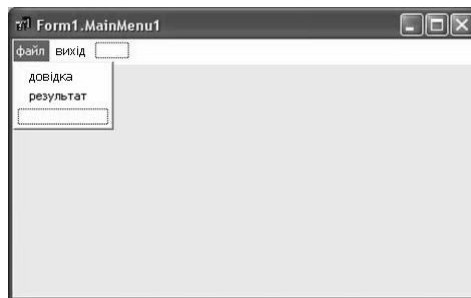


Рис. 2 Створення меню

8. Дизайн практично завершено. Зберігаємо та компілюємо проект (рис. 3).

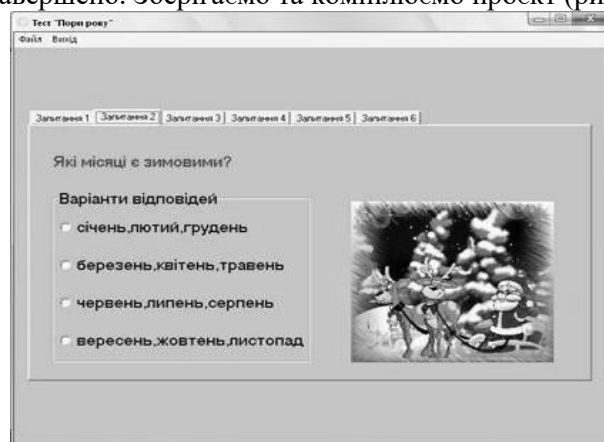


Рис. 3 Дизайн проекту

9. Для кнопки *Вихід* головного меню програмуємо процедуру *close* для виходу з програми.
10. Створюємо форму для довідки *NewForm*. Змінюємо її назву на *Довідка*. На формі надрукуємо інструкцію через компонент *Label*, також вставляємо картинку (рис. 4).

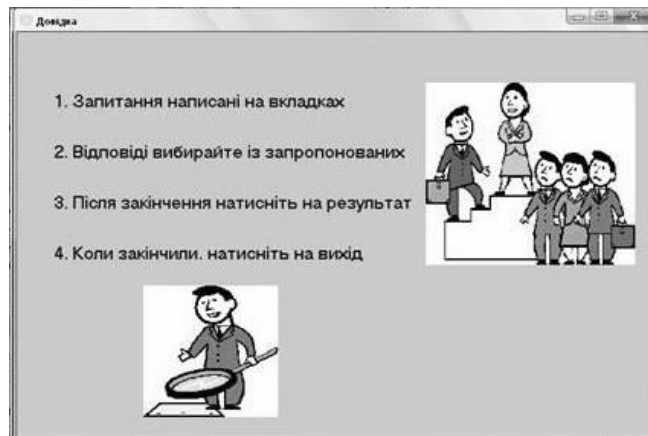


Рис. 4 Форма довідки

11. Створимо процедуру, яка при натисканні на пункт *Допомога* головного меню буде відкривати вікно з довідкою: *Form2.Showmodal*.
12. Для того, щоб використовувати при роботі програми другу форму, в Unit1 добавляємо uses Unit2.
13. Підрахунок балів має здійснюватись при натисканні на пункт *Результат*. При цьому потрібно передбачити необхідність вибору хоча б одного варіанту з запропонованих, наприклад, виведення повідомлення:

```
if RadioGroup1.ItemIndex=-1  
then ShowMessage ('Потрібно вибрати відповідь');
```

Для 1 питання зробити процедуру для кнопки *OnClick*, в якій прописати фрагмент:

```
If StrToInt(Edit1.text)=4 then ball:=ball+1;
```

Аналогічно 5 запитання.

14. Передбачити виведення повідомлення з оцінкою за 12-бальною шкалою оцінювання.
Фрагмент лістингу програми:

```
interface  
uses  
Unit2, Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, ComCtrls, StdCtrls;  
type  
{ TForm1 }  
TForm1 = class(TForm)  
Button1: TButton;  
PageControl1: TPageControl;  
TabSheet1: TTabSheet;  
procedure Button1Click(Sender: TObject);  
private  
{ private declarations }  
public  
{ public declarations }  
end;  
var  
Form1: TForm1;  
Ball: integer;  
  
implementation  
{ $R *.lfm }  
{ TForm1 }  
  
procedure TForm1.N4Click(Sender: TObject);  
begin
```

```

close;
end;
procedure TForm1.N2Click(Sender: TObject);
begin
  Form2.ShowModal;
end;
procedure TForm1.N3Click(Sender: TObject);
begin
  Ball:=0;
  if RadioGroup2.ItemIndex=-1 then ShowMessage ('Потрібно вибрати відповідь') else
  if RadioGroup2.ItemIndex=0 then Ball:=Ball+1;
  if RadioGroup3.ItemIndex=-1 then ShowMessage ('Потрібно вибрати відповідь') else
  if RadioGroup3.ItemIndex=1 then Ball:=Ball+1;
  if RadioGroup4.ItemIndex=-1 then ShowMessage ('Потрібно вибрати відповідь') else
  if RadioGroup4.ItemIndex=3 then Ball:=Ball+1;
  if RadioGroup6.ItemIndex=-1 then ShowMessage ('Потрібно вибрати відповідь') else
  if RadioGroup6.ItemIndex=1 then Ball:=Ball+1;
  ShowMessage ('Ваш результат'+IntToStr(Ball));
end;
end.

```

Висновки та перспективи подальшого дослідження. У роботі розкрито особливості розробки додатків для перевірки знань учнів із використанням мови програмування Lazarus. Розроблені додатки у середовищі програмування Lazarus мають як перспективи, так і недоліки. Такі додатки можуть бути використані як приклад показу можливості створювати контрольні програми для учнів школи будь-якого віку та з будь-якого предмета. Недоліком таких додатків є те, що вони «одноразові», тобто створивши будь-який тест, не можна його корегувати, удосконалювати, додавати чи змінювати запитання чи відповіді. Але з іншого боку, зробивши один, два чи більше тестів, запам'ятавши програмний код та навчившись створювати форму, на створення такого тесту витрачається 15-30 хвилин.

Список бібліографічного опису

1. Алексеев С. Р. Самовчитель з програмування на Free Pascal та Lazarus. Донецьк : ДонНТУ, Технопарк ДонНТУ УНІТЕХ, 2011. 503 с.
2. Бублик В.В. Об'єктно-орієнтоване програмування. К.: ІТкнига, 2015. 624 с.
3. Гуриков С. Р. Програмування в середовищі Lazarus для школярів та студентів : навч. посібник. К. : Форум : НДЦ ІНФР, 2016. 336 с.
4. Белов В. В. Програмування в Delphi : процедурне, об'єктивно орієнтоване, візуальне: Навч. посібник для вузів. К.: Гаряча лінія, Телеком, 2014. 240 с.
5. Мансуров К. Т. Основи програмування у середовищі Lazarus. К.: Palmarium Academic Publishing, 2013. 772 с.
6. Завадський І. О., Заболотний Р. І. Основи візуального програмування. К.: Вид. група ВHV. 2007. 272 с.
7. Семакін І. Н. Основи програмування : підручник. К.: Вища школа, 2002. 432 с.
8. Информатика : підруч. для 9-го кл. загальноосвіт. Навч. закл. Київ. : Генеза, 2017. 288 с.
9. URL: <https://naurok.com.ua/stattya-vikoristannya-suchasnih-testovih-tehnologiy-dlya-perevirki-znan-uchniv-naurokah-273884.html> (дата звернення: 10.12. 2022)

References

1. Alekseev E.R. Free Pascal and Lazarus programming tutorial. Donetsk: DonNTU, Technopark of DonNTU UNITECH, 2011. 503 p.
2. Bubyk V.V. Object-oriented programming. K.: IT book, 2015. 624 p.
3. Gurikov S. R. Programming in the Lazarus environment for schoolchildren and students: teaching. manual. K.: Forum: NDC INFR, 2016. 336 p.
4. Belov V.V. Programming in Delphi: procedural, objective-oriented, visual: Education. a guide for universities. K.: Hotline, Telecom, 2014. 240 p.
5. Mansurov K. T. Fundamentals of programming in the Lazarus environment. K.: Palmarium Academic Publishing, 2013. 772 p.
6. Zavadskiy I.O., Zabolotnyi R.I. Fundamentals of visual programming K.: Ed. BHV group. 2007. 272 p.
7. Semakin I. N. Fundamentals of programming: a textbook. K.: Higher School, 2002. 432 p.
8. Informatics: tutor. for the 9th grade general education Education closing Kyiv. : Geneza, 2017. 288 p.
9. URL: <https://naurok.com.ua/stattya-vikoristannya-suchasnih-testovih-tehnologiy-dlya-perevirki-znan-uchniv-naurokah-273884.html> (date of application: 10.12. 2022)