

УДК 373.5.016:004

## РЕАЛІЗАЦІЯ МОЖЛИВОСТЕЙ ІНТЕГРОВАНОГО СЕРЕДОВИЩА РОЗРОБКИ LAZARUS У ВИВЧЕННІ ІНФОРМАТИКИ В ШКОЛІ

**Васенко Олександр Васильович,**

доцент ДВНЗ «Переяслав-Хмельницький державний педагогічний університет імені Григорія Сковороди», доцент кафедри математики, інформатики та методики навчання, кандидат історичних наук,  
vasenko.olexandr@gmail.com.



**Анотація.** У статті здійснена спроба узагальнення теоретичних відомостей про інтегроване середовище розробки Lazarus, розглянуто його характерні можливості та переваги порівняно з подібними інструментальними програмними засобами, а також проаналізовано перспективи його використання в процесі викладання шкільного курсу інформатики.

**Ключові слова:** інтегроване середовище розробки, Lazarus, об'єктно-орієнтоване програмування, кросплатформна система, курс інформатики, інтерфейс.

Сучасна тенденція розвитку суспільства та впровадження інформаційно-комунікаційних технологій практично в усі сфери людської діяльності створила умови для ревізії поглядів на модель підготовки учнів загальноосвітніх навчальних закладів. Особливо актуально це стосується вивчення курсу інформатики, де згідно методичних рекомендацій в 2016–2017 навчальному році пріоритетним напрямком у навчально-виховному процесі стає компетентісно-орієнтований підхід. Згідно якого передбачається використання відповідних компетентісних завдань і задач протягом всього навчального курсу інформатики, а також перехід до використання подійно- та об'єктно-орієнтованого програмування, починаючи з 8 класу. Це, у свою чергу, зумовило перелік рекомендованих до вивчення мов програмування — ObjectPascal, VisualBasic, Python, Java, C#, C++ тощо і відповідно вибір середовищ програмування — Lazarus (мова ObjectPascal), VisualStudio (безкоштовна версія CommunityEdition, мова VisualBasic), IDLE forPython (мова Python) та інші [1].

**Аналіз останніх досліджень і публікацій.** Сучасний стан дослідження проблеми використання різних мов і середовищ програмування висвітлено у дослідженнях М. І. Жалдака, Н. В. Морзе, С. А. Ракова, Ю. С. Рамського, С. О. Семерікова, О. М. Спіріна та інших. Стосовно висвітлення питання використання середовища програмування Lazarus в навчальному процесі загальноосвітньої школи, то сучасні дослідження, в основному, містять фрагментарні відомості і стосуються окремих аспектів або методичних рекомендацій.

З огляду на це, метою статті є узагальнення відомостей та аналіз характерних можливостей середовища програмування Lazarus як перспективної платформи для застосування у вивченні курсу інформатики в сучасній загальноосвітній школі.

Розглядаючи сучасні середовища програмування, запропоновані для викладання курсу інформатики у школі, слід зазначити, що вони містять, у цілому, схожий набір засобів і компонентів для використання в навчальному процесі. З огляду на це, на нашу думку, перспективним є впровадження в навчальний процес загальноосвітньої школи середовища програ-

мування Lazarus. Перевагами такого кроку, насамперед є наявність значного обсягу україномовної літератури з паскалеподібних мов орієнтованої на використання в навчальному процесі шкіл та педагогічних університетів. Також слід зазначити широке використання мови Pascal під час навчання алгоритмізації та програмування у школах, як порівняно простої для засвоєння для початківців, і відповідно проведення шкільних олімпіад з її використанням.

Ще однією перевагою є те, що середовище програмування Lazarus є безкоштовним і вільно поширюваним на основі ліцензій GNU General Public License (GNU GPL) і GNU Lesser General Public License (GNU LGPL), що є важливим, особливо для використання в державних освітніх установах, дозволяючи закладам освіти уникнути використання піратського програмного забезпечення.

Крім того, Lazarus є системою з відкритим вхідним кодом, базою для якого є компілятор FreePascal Compiler з додаванням інтегрованого середовища розробки (IDE—Integrated Development Environment). Вбудовані в Lazarus компоненти та бібліотеки дозволяють використовувати це середовище програмування для створення додатків, що містять графічний інтерфейс (GUI — Graphical User Interface).

Також варто зазначити, що крім графічного інтерфейсу, Lazarus, базується на кросплатформній бібліотеці візуальних і не візуальних компонентів LCL (Lazarus Component Library), що є сумісними з VCL Delphi. Тому Lazarus може працювати під керівництвом різних операційних систем, таких як Linux, MacOSX, UNIX-подібних, Windows, Android. Завдяки цьому, додатки, що створені для однієї операційної системи можна перекомпілювати підіншу. Тобто використовуючи середовище Lazarus, можна створювати кросплатформні додатки [2].

Важливою рисою, що відрізняє Lazarus від більшості середовищ програмування свого класу, є наявність локалізації українською мовою. Слід сказати, що розглянуті у змістовій лінії «Моделювання, алгоритмізація та програмування» питання шкільного курсу інформатики є досить складними для засвоєння значною частиною учнів. Причинами цього були в

більшості випадків — необхідність дотримання жорстких правил запису програмного коду та здійснення перекладу помилок компіляції з технічної англійської мови. Відтак, україномовний інтерфейс середовища Lazarus поряд з розумінням відлагоджувальної інформації значною мірою дозволяє вирішити вказані проблеми у вивченні програмування учнями.

Також серед основних характеристик середовища програмування Lazarus слід відзначити:

- наявність редактора коду, із системою підказок, гіпертекстовою навігацією та автозавершенням коду;
- наявний редактор форм та інспектор об'єктів дуже схожі на відповідні елементи Delphi;
- близькість набору елементів керування бібліотеки візуальних компонентів LCL Lazarus (Lazarus Component Library) та VCL (Visual Component Library) Delphi;
- наявність вбудованого відлагоджувача;
- підтримка різних типів синтаксису Pascal: ObjectPascal, TurboPascal, MacPascal, Delphi (підтримуються з боку компілятора);
- повністю Юнікодний (UTF-8) інтерфейс і редактор, і тому відсутні проблеми з портуванням коду, що містить національні символи;
- наявний власний формат керування пакетами даних;
- підтримувані для компіляції Linux, Microsoft Windows (Win32, Win64), Mac OS X, FreeBSD, WinCE, OS/2;
- середовище функціонування — операційні системи Linux, Microsoft Windows (Win32, Win64), Mac OS X, FreeBSD, WinCE, OS/2[3].

Вважаємо за доцільне детальніше проаналізувати інтерфейс середовища розробки Lazarus, як основного елемента з яким постійно контактує користувач, а також розглянути можливості, що він надає при виконанні поставлених завдань в рамках навчального процесу. Відразу слід відзначити, що інтерфейс Lazarus дуже схожий на інтерфейс Delphi, а це до-

зволяє швидше адаптуватися користувачам, які працювали з цією платформою. Елементами, що створюють інтерфейс Lazarus, є — головне вікно програми з палітрою компонентів, вікно інспектора об'єктів, редактор коду та ін. (рис. 1).

**Головне вікно** — виконує основні функції керування проектом створеної програми. Основними функціональними одиницями цього вікна є три блоки: головне меню — з командами керування середовищем і компіляцією, панель інструментів — забезпечує швидкий доступ до основних команд головного меню, палітра компонентів — надає доступ до основних компонентів середовища розробки.

**Вікно інспектора об'єктів (ObjectInspector)** — тут розміщені параметри активного компонента, розташованого на формі, які програміст може редагувати в процесі розробки програми. У верхній частині знаходиться область дерева об'єктів, тобто відображається ієрархічна структура компонентів. Основну частину інспектора об'єктів займають чотири закладки: **Properties (Властивості)** — містить властивості компонента **Events (Події)** — містить події, які може обробляти активний компонент, **Favorites (Вибране)** — містить вибрані властивості і події, **Restricted (Обмеження)** — містить обмеження властивостей і подій активного компонента відносно операційної системи, що використовується.

**Вікно редактора коду програми (SourceEditor)** — основним призначенням є введення і редагування тексту програми. Багато функцій цього редактора співпадають з можливостями звичайних текстових редакторів і включають в себе — виділення, копіювання, вставку, виконання пошуку і заміну фрагментів тексту. Поряд з цим у ньому наявна низка можливостей, що допомагає в розробці програм. У ньому, зокрема, є функція підсвічування синтаксису, і не лише мови Pascal. Для забезпечення структурування коду з метою полегшення можливостей його читання наявна функція зсуву фра-

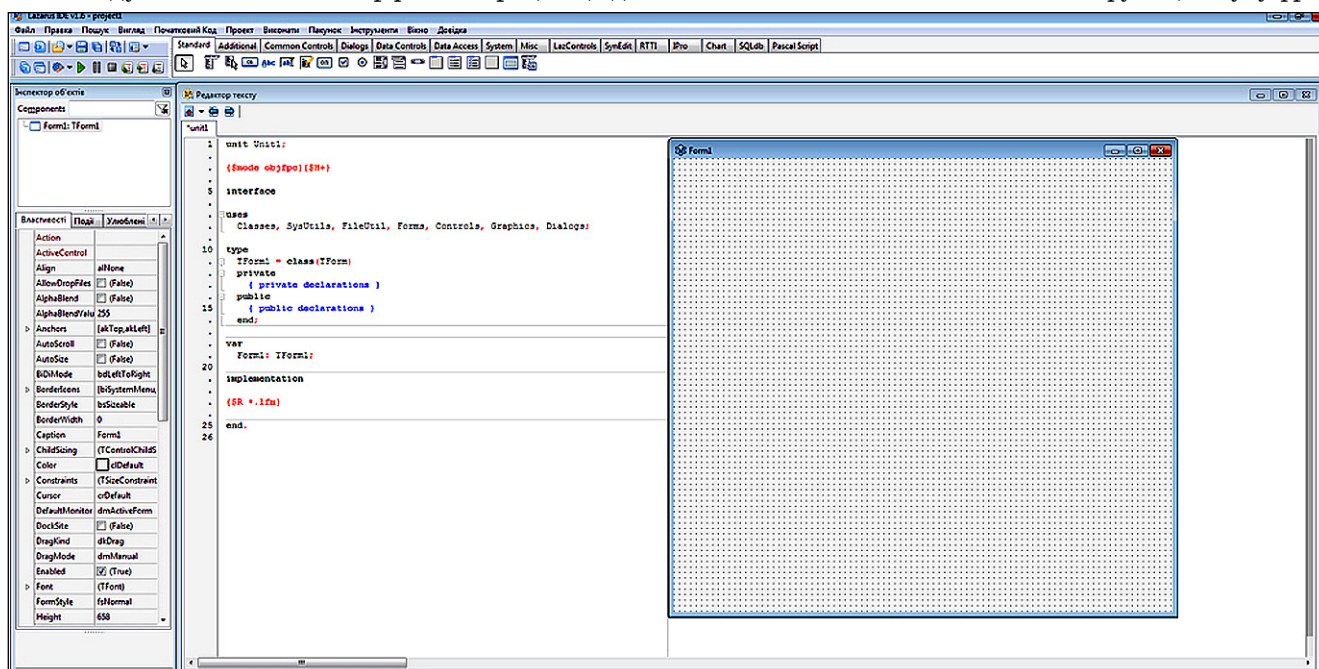


Рис. 1. IDE Lazarus у Windows

гментів тексту вліво або вправо. Пошук в редакторі, крім звичайних функцій, допомагає у визначенні місця, де було здійснено оголошення ідентифікатора. Ця інформація надається за допомогою контекстної підказки, що з'являється у разі наведення курсору миші на відповідний ідентифікатор. Також є інше використання цього пошуку, утримуючи кнопку **Ctrl** і, натиснувши ліву кнопку миші на підключеному модулі, компоненті, властивості або методи відбудеться перехід до відповідного модуля або на відповідний клас компонента. Також якщо переміститися на оголошення деякого методу і натиснути комбінацію кнопок **Ctrl+Shift+стрілка вниз**, то відбудеться перехід на визначення цього методу, а для переходу з деякого метода до місця його оголошення використовується комбінація кнопок **Ctrl+Shift+стрілка вгору**.

У середовищі розробки Lazarus завершення ідентифікаторів або завершення коду, що дозволяє вибрати об'єкт, властивість або метод зі списку за введеними першими літерами. Активація списку з переліком можливих значень, відбувається шляхом натискання комбінації кнопок **Ctrl+Space**, якщо введена кількість символів буде відповідати лише одному можливому варіанту, то він автоматично додасться до редактора коду.

Підказка параметрів дає змогу переглянути список необхідних параметрів підпрограми (процедури чи функції) у спливаючому вікні. Вона викликається комбінацією кнопок **Ctrl+Shift+Space** в області опису фактичних параметрів підпрограми. У цьому вікні міститься опис відповідної підпрограми з її формальними параметрами, а формальний параметр, для якого в цей час повинно відбуватися введення фактичного параметра, буде підсвічено жирним шрифтом.

Опція завершення класів використовується для автоматичного генерування завершення коду, вона активується натисканням комбінації кнопок **Ctrl+Shift+C**. Тобто, після оголошення деякого методу в класі і застосування вказаної комбінації кнопок, то в розділі реалізації модуля буде згенеровано каркас відповідного методу [4].

**Вікно форми** — проект вікна майбутньої форми. Тут відбувається проектування та налаштування зовнішнього вигляду програми, шляхом розміщення елементів управління. Варто зауважити, що в середовищі розробки Lazarus порівняно з Delphi, елементи сторінок розробки Standard, Additional, Dialogs практично ідентичні.

**Вікно повідомлень (Messages)** — призначене для виведення повідомлень компілятора, відображає хід успішної компіляції чи перелік знайдених помилок [5].

Поряд з компонентами для створення графічної складової програм, середовище розробки Lazarus містить інструменти для створення консольних додатків. Особливістю роботи в цьому режимі, порівняно з тим же Delphi, є можливість налаштування окремих характеристик та ввімкнути додаткові можливості в новому проекті (рис. 2).

Зокрема, **Usage message (-h options)** — підключає процедуру отримання повідомлень довідки (це повідомлення виводиться під час запуску додатка з параметром «h» або «help»).

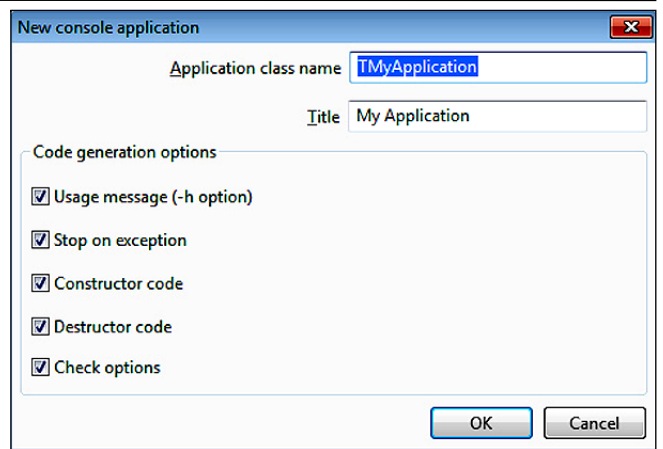


Рис 2. Допоміжне вікно створення нового консольного проекту

**Stoponexception** — вмикає властивості StopOnException, що дозволяє зупинити роботу додатка при виникненні виключної ситуації.

**Constructorcode** — додає процедуру конструктора проекту.

**Destructorcode** — додає процедуру деструктора проекту.

**Checkoptions** — додає блок перевірки введених параметрів на відповідність тим, що можуть бути опрацьовані додатком [6].

Також слід відмітити, що в результаті компіляції навіть порожньої форми в середовищі розробки Lazarus, розмір виконуваного файлу буде досить великим порівняно з подібними файлами в інших середовищах розробки. Розробники пояснюють це тим, що бінарні файли містять дані для відлагодження, які використовуються програмою GDB (GNU Debugger). Проте, можна здійснити налаштування компілятора для автоматичного видалення цих даних. Для цього необхідно скористатися пунктом головного меню **Проект/Параметри** проекту, у категорії **Compilation and Linking** необхідно відмітити пункти **Розумна компіляція (-CX)** (Підтримка «розумного компонування») та **Розумне компонування (-XX)** (рис. 3), і у категорії **Debugging** зняти відмітку з пункту **Створити інформацію для GDB** (повільніше і збільшує розмір exe) (рис. 4).

У новіших версіях Lazarus, процес зменшення виконуваного файлу значно спростилося. І для цього необхідно в головному меню перейти в **Параметри про-**

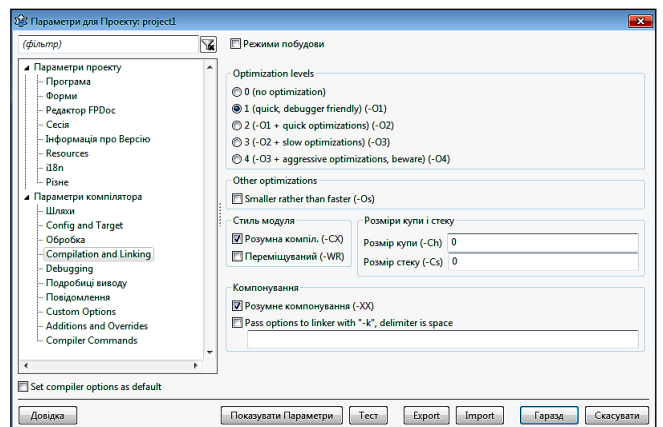


Рис 3. Категорія **Compilation and Linking** вікна налагодження параметрів проекту

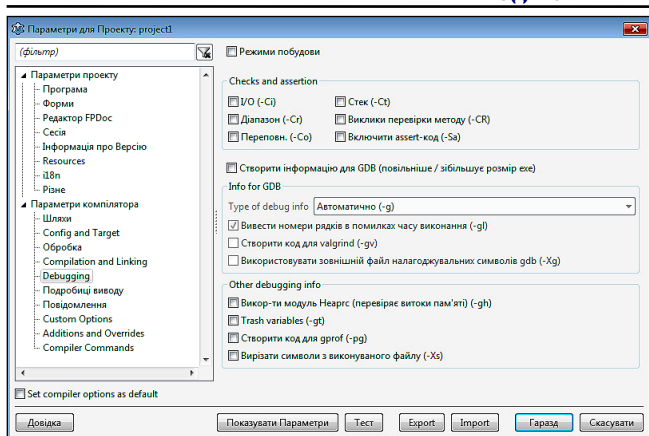


Рис 4. Категорія Debugging вікна налагодження параметрів проекту

екту (Проект/Параметри проекту) і перейти до категорії Параметри компілятора. Ставимо відмітку пункту Режим побудови, а далі натискаємо кнопку, що з'явилася поряд (рис. 5). Тут натискаємо кнопку Create Debug and Release modes, далі у вікні з'явиться ще два режими побудови проекту — Release та Debug (рис. 6). Саме режим Release створює маленькі виконувані файли, а кнопка для вибору необхідного режиму буде додано на панель інструментів.

Аналогічні дії можна зробити вже з відкомпільованим проектом, скориставшись утилітою, що міститься в каталозі з встановленим Lazarus. Для цього необхідно виконати команду strip — strip-all<ім'я виконаного файла>.

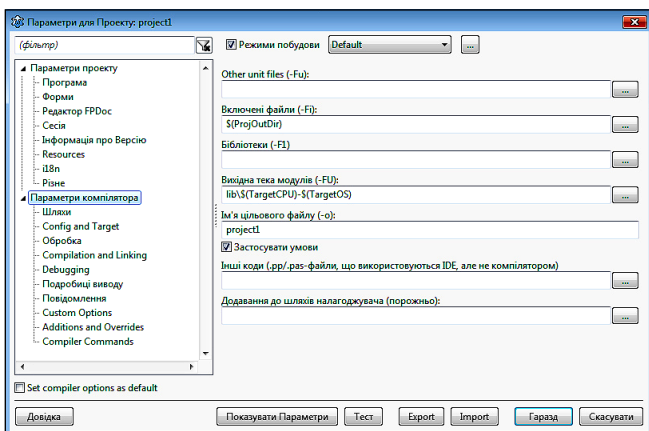


Рис 5. Категорія Параметри компілятора вікна налагодження параметрів проекту

Вказані особливості інтегрованого середовища розробки Lazarus, не відіграють особливої ролі під час його використання в навчальному процесі, але мають значення у разі створення складніших проектів. При цьому важливою властивістю середовища розробки Lazarus є можливість додавання до нього нових компонентів. Для цього потрібно перекомпілювати всю візуальну систему. Це пов'язано з тим, що такий компонент тісно інтегрується із середовищем програмування і стає його частиною.

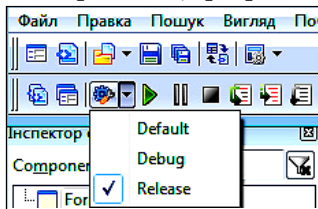


Рис 6. Кнопка вибору параметрів проекту на панелі інструментів

Відзначимо, що враховуючи всі перелічені характеристики та можливості середовища розробки Lazarus його може використовуватися не лише як середовище для навчання програмування, а і як потужний інструмент для створення програмних засобів навчального призначення. Цьому сприяють як технологічні можливості середовища програмування, так і доступність літератури з розробки коду програми, а також ліцензійні умови використання Lazarus.

**Висновки.** У підсумку варто зазначити, що інтегроване середовище розробки Lazarus має необхідний технічний набір інструментів для використання його в навчальному процесі загальноосвітньої школи. Ключовими перевагами його порівняно з подібними за класом середовищами є наявність україномовної локалізації та вільне поширення, що дозволяє зняти проблему використання нелегального програмного забезпечення у закладах освіти. Крім того Lazarus, може застосовуватися і для створення прикладного програмного забезпечення з графічним інтерфейсом, що значно розширює потенційні можливості вчителя по використанню інформаційно-комунікаційних технологій у навчальному процесі.

\* \* \*

**Васенко А. В. Реалізація можливостей інтегрованої середовища розробки Lazarus в процесі вивчення інформатики в школі**

**Анотація.** В статті зроблена спроба обобщення теоретических свідень о інтегрованої середовища розробки Lazarus, рассмотрены её характерные возможности и преимущества в сравнении с подобными инструментальными программными средствами, а также проанализированы перспективы её использования в процессе преподавания школьного курса информатики.

**Ключевые слова:** интегрированная среда разработки, Lazarus, объектно-ориентированное программирование, кроссплатформенная система, курс информатики, интерфейс.

\* \* \*

**Vasenko Aleksandr. Implementation of opportunities of Lazarus the study computer science in school**

**Abstract:** The attempted synthesis of theoretical information about integrated development environment Lazarus, considered its characteristic features and advantages over similar tool software, and analyzed the prospects for its use in teaching computer science school course.

**Keywords:** integrated development environment, the Lazarus, object-oriented programming, cross-platform system, computer science course interface.

## Література

1. Додаток до листа Міністерства освіти і науки України від 17.08.2016р. №1/9-437. [Електронний ресурс]. — Режим доступу: <http://old.mon.gov.ua/ua/about-ministry/normative/6119>.
2. Overview of Free Pascal and Lazarus [Електронний ресурс]. — Режим доступу: [http://wiki.lazarus.freepascal.org/Overview\\_of\\_Free\\_Pascal\\_and\\_Lazarus](http://wiki.lazarus.freepascal.org/Overview_of_Free_Pascal_and_Lazarus).
3. Харченко В. М. До питання використання Lazarus при вивченні шкільного курсу інформатики / Харченко В. М., Харченко М. В. // FOSSLviv 2014, 24–27 квітня 2014 р. — Л., 2014 — С. 43–45.
4. Мансуров К. Т. Основы программирования в среде Lazarus / К. Т. Мансуров, 2010. — 772 с.
5. Алексеев Е. Р. Самоучитель по программированию на Free-Pascal и Lazarus / Алексеев Е. Р., Чеснокова О. В., Кучер Т. В. — Донецк.: ДонНТУ, Технопарк ДонНТУ УНИТЕХ, 2009. — 503 с.
6. Використання ВПЗ у процесі вивчення основ програмування / Юрій Горошко, Андрій Костюченко, Микола Шкардибарда // Інформатика та інформаційні технології. — 2012. — №1. — С. 22–25.