

УДК 378.016:004.41'23

# ВИКОРИСТАННЯ ВПЗ У ПРОЦЕСІ ВИВЧЕННЯ ОСНОВ ПРОГРАМУВАННЯ

**Юрій ГОРОШКО**, кандидат педагогічних наук, доцент кафедри інформатики та обчислювальної техніки Чернігівського національного педагогічного університету імені Т. Г. Шевченка;

**Андрій КОСТЮЧЕНКО**, асистент кафедри інформатики та обчислювальної техніки Чернігівського НПУ імені Т. Г. Шевченка;

**Микола ШКАРДИБАРДА**, асистент кафедри інформатики та обчислювальної техніки Чернігівського НПУ імені Т. Г. Шевченка

**Анотація.** У статті розглядаються особливості застосування ВПЗ у процесі викладання програмування на фізико-математичному факультеті ЧНПУ імені Т. Г. Шевченка. Основна увага приділяється головним аспектам використання вільно поширюваних середовищ програмування Free Pascal та Lazarus. Обґрунтовано доцільність їх застосовуваних в навчальному процесі педагогічного ВНЗ та школи, їх переваги та недоліки. Вказано ключові моменти, які треба враховувати під час використання цих програм. Особливості середовища Lazarus порівнюються з середовищем Delphi.  
**Ключові слова:** ВПЗ, програмування, Lazarus, Free Pascal, Delphi.

**Аннотация.** В статье рассматриваются особенности применения СПО в процессе преподавания программирования на физико-математическом факультете ЧНПУ имени Т. Г. Шевченко. Основное внимание уделено ключевым аспектам использования свободно распространяемых сред программирования Free Pascal и Lazarus. Обоснованно целесообразность их использования в учебном процессе педагогического вуза и школы, их преимущества и недостатки. Указаны ключевые моменты, которые нужно учитывать при использовании этих программ. Особенности среды Lazarus сравниваются со средой Delphi.

**Ключевые слова:** СПО, программирование, Lazarus, Free Pascal, Delphi.

**Summary.** The issues of using of Free Software in teaching of programming on the Physics and Mathematics department of Shevchenko Chernihiv National Pedagogical University are described in the article. The main attention is paid to the features of the use of such open source programming environments as Free Pascal and Lazarus. The article contains the reasoning for the expediency of their use in the educational process of pedagogical universities and schools, their advantages and disadvantages. The article deals with key points of using these programs. The features of Lazarus environment are compared with the environment of Delphi.

**Key words:** Free Software, programming, Lazarus, Free Pascal, Delphi.

З огляду на плановий перехід до застосування ВПЗ на фізико-математичному факультеті ЧНПУ імені Т. Г. Шевченка, для вивчення основ алгоритмізації було обрано середовище Free Pascal, а для вивчення основ ООП – Lazarus. Вибір цих середовищ програмування зумовлений наявністю великого обсягу літератури з паскалеподібних мов програмування для шкіл і педагогічних університетів, а також порівняною легкістю їх засвоєння для непрофесійних програмістів. Важливо, що Free Pascal є однією з основних мов на олімпіадах з інформатики для школярів в Україні та світі.

Як вже зазначалось, середовище Free Pascal застосовується тільки для вивчення основ алгоритмізації, тому говорити про особливості його використання немає сенсу. Єдине, що хотілося б зазначити, це те, що в ньому використовується діалект Free Pascal, в якому тип Integer є 16-бітним, тому бажано на першому ж занятті обрати ін-

ший діалект, наприклад Object Pascal. Для розв'язування олімпіадних задач потрібно ознайомити студентів із знаковими та беззнаковими 64-бітними типами даних.

Розглянемо особливості середовища Lazarus у порівнянні з середовищем Delphi. Подібно до Delphi, Lazarus є середовищем RAD (з англ. *rapid application development* – швидка розробка додатків), що дає змогу швидко створювати інтерфейс користувача. На відміну від Delphi Lazarus є кросплатформним додатком, що підтримує такі ОС, як GNU/Linux, Microsoft Windows, Mac OS X, FreeBSD, WinCE.

**Інтерфейс.** Інтерфейс Lazarus практично повністю повторює інтерфейс Delphi. Таке саме вікно редактора, той самий інспектор об'єктів, однаково головне вікно програми з палітрою компонентів. Тому перехід з Delphi на Lazarus не викликає психологічного бар'єра для його використання. Крім

того, в Lazarus є можливість обрати іншу мову інтерфейсу, що сприяє позитивному ставленню студентів до продукту.

Щодо вигляду палітри компонентів, Lazarus нагадує Delphi 5. Елементи сторінок Standard, Additional, Dialogs практично ідентичні. Сторінка Standard відрізняється лише відсутністю компонентів для організації фреймів (рис. 1).

Сторінки Common Controls та Misc частково відповідають сторінці Win32 в Delphi: є компоненти для організації дерев, списків, Page і Tab control, смуг прокрутки, рядків стану тощо. Тому щодо стандартних елементів управління Lazarus не поступається Delphi. Враховуючи темпи розвитку відкритого ПЗ, не варто перейматися, якщо важко знайти певний компонент, можна відшукати його аналог в Інтернеті.

Так, компанія Nevrona портувала під Lazarus весь пакет Indy, силами ентузіастів був написаний пакет FBL, який являє собою аналогічну функціональність FIB-ам (Fast InterBase — бібліотека компонентів для прямого доступу до InterBase та Firebird) тощо. Звичайно, користувачам платних (закритих) бібліотек під час портування проектів може бути досить складно, проте наявних компонентів цілком достатньо для викладання основ ООП.

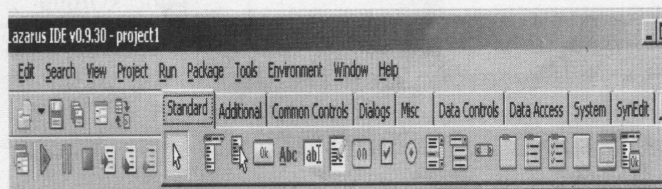


Рис. 1. Палітра компонентів Standard в Lazarus

**Особливості синтаксису.** Як кросплатформений компілятор Free Pascal підтримує декілька діалектів мови, зокрема FreePascal, Delphi, Turbo Pascal, Object Pascal (за замовчуванням) та Mac Pascal. Для перенесення проектів, написаних з використанням Delphi, необхідно виставити відповідний діалект у налагодженнях або просто виправити директиву компілятора на початку кожного модуля, тобто замість вказівки за замовчуванням `{ $mode objfpc } { $H+ }` вказати `{ $mode delphi } { $H+ }`.

**Особливості консольних додатків.** Процес створення нового консольного додатка в середовищі Lazarus є аналогічним до його створення в середовищі Delphi, проте у Lazarus пропонується встановити певні характеристики та включити додаткові мож-

ливості в тіло нового проекту (рис. 2):

- *Usage message (-h options)* – додає процедуру для виведення повідомлення про допомогу (це повідомлення буде виводитись під час запуску цього додатка з параметром “h” або “help”);
- *Stop on exception* – включення властивості StopOnException, в результаті чого при виникненні виключної ситуації виконання додатку буде зупинено;
- *Constructor code* – додає процедуру конструктора проекту;
- *Destructor code* – додає процедуру деструктора проекту;
- *Check options* – додає блок перевірки введених параметрів на відповідність тим, що можуть бути опрацьовані додатком.

Навіть якщо вимкнути всі перемикачі та не включати зазначені додаткові можливості, то консольний додаток Lazarus, за замовчуванням, буде набагато більшим від консольного додатка Delphi. Він частково нагадуватиме проект, в якому визначено клас для додатка, зміну цього класу, процедуру DoRun для виконання операторів користувача, а тіло програми міститиме оператори з виділення пам'яті для додатка, запуску процедури DoRun та очищення виділеної пам'яті.

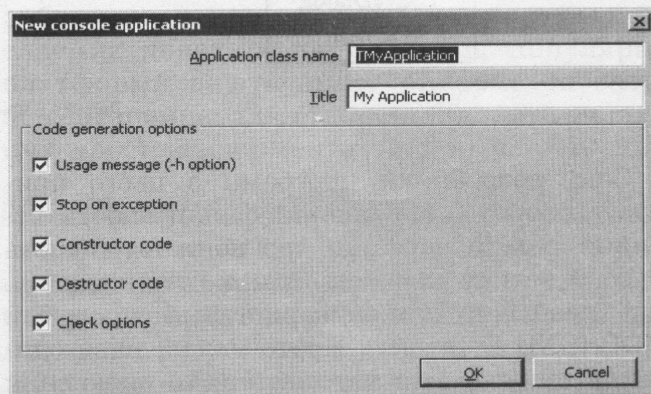


Рис. 2. Допоміжне вікно створення нового консольного проекту

**Бібліотека класів.** Бібліотека класів, відома як LCL (Lazarus Component Library), є кросплатформеною та сумісною з VCL Delphi. Її можна подати у вигляді деревоподібної структури (рис. 3).

Можна побачити, що бібліотека класів LCL має низку додаткових проміжних класів, проте основна структура дерева залишається аналогічною.

**Розмір виконуваного файлу.** У початківця може виникнути невеликий шок після компіляції порожньої форми – розмір виконуваного файлу займатиме понад 10 Мб.



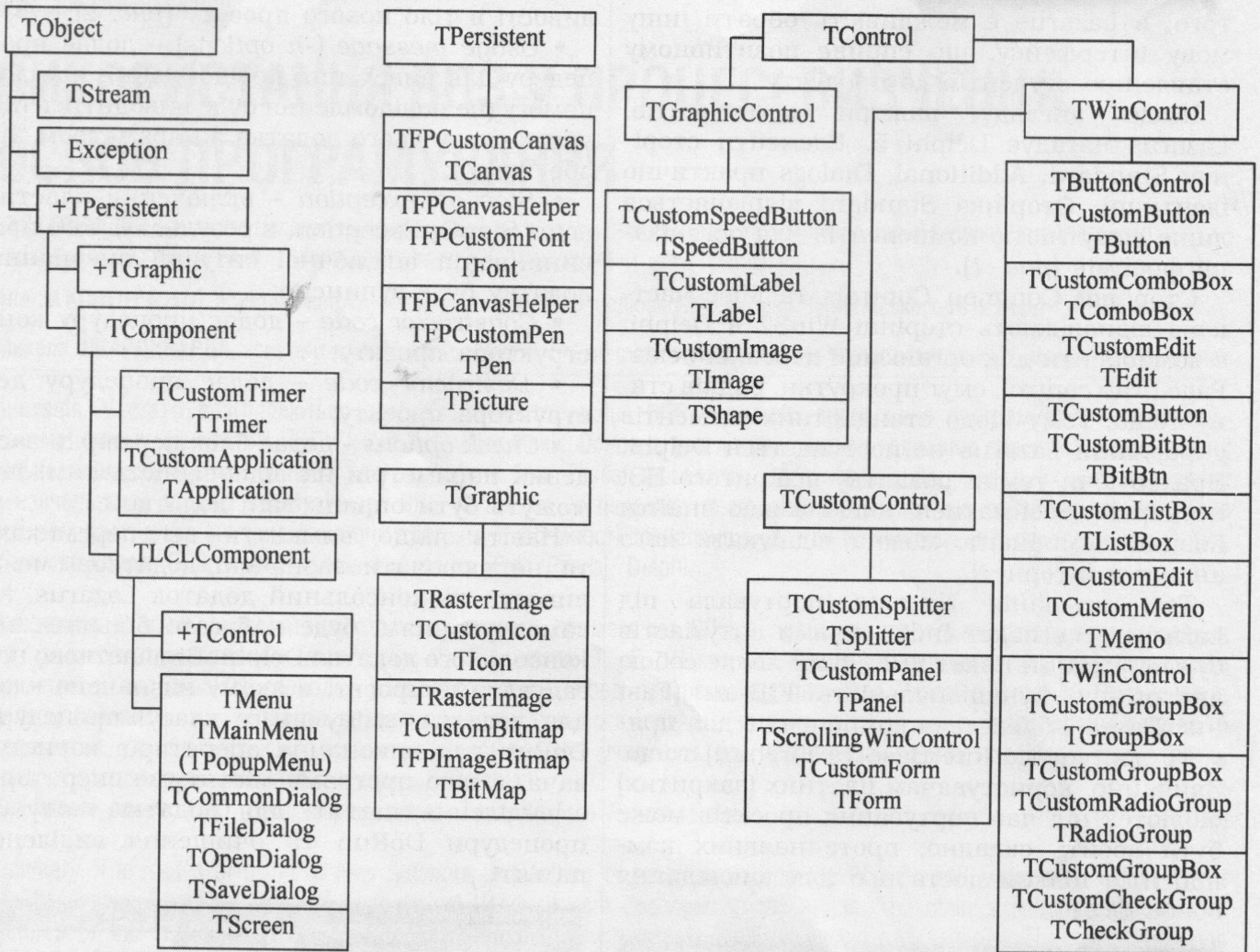


Рис. 3. Бібліотека класів LCL

Самі розробники програми з цього приводу говорять, що бінарні файли мають великий розмір через те, що вони включають в себе велику кількість даних для відладки, які необхідні для використання програми GDB (GNU Debugger). Проте можна налагодити компілятор для автоматичного вилучення даних відладки з файлів, що виконуються. Для цього необхідно скористатися пунктом головного меню *Project/ Project options* (Проект/ Параметри проекту). У категорії *Code generation* (Генерування коду) необхідно відмітити пункт *Smart Linkable (-CX)* (Підтримка «розумного» компонування) (рис. 4), а в категорії *Linking* (Зв'язування) – пункти *Strip Symbols From Executable (-Xs)* (Вилучити дані для відладки з виконуваного файла), *Link Smart (-XX)* («Розумне» компонування) і зняти відмітку з пункту *Display Line Numbers in Run-time error Backtraces (-g)* (Виводити номери рядків в повідомленнях про помилки часу виконання) (рис. 5). Варто зауважити, що вимкнення останнього пункту призведе до неможливості використовувати «Точки зупинки» при відладці.

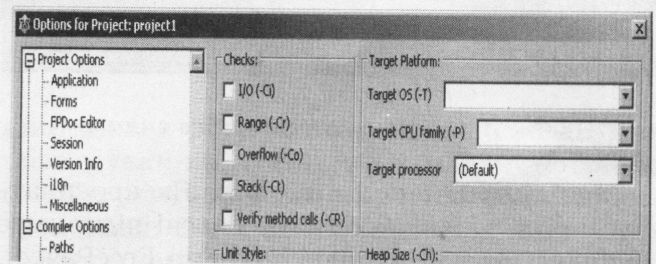


Рис. 4. Категорія Code вікна налагодження параметрів проекту

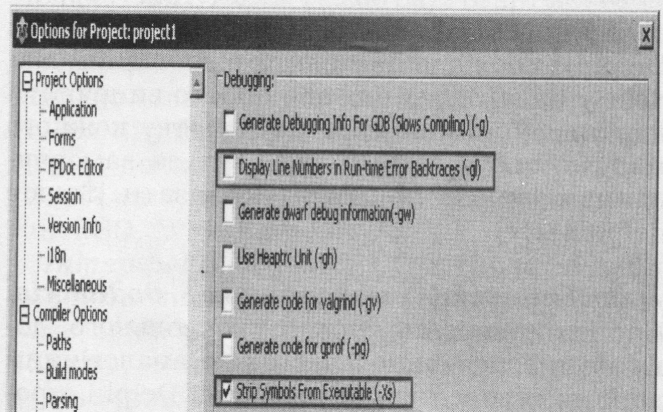


Рис. 5. Категорія Linking вікна налагодження параметрів проекту

Те ж саме можна зробити вже з відкомпільованим проектом, скориставшись утилітою *strip*, що міститься в каталозі із встановленим середовищем Lazarus. Для цього необхідно виконати команду *strip --strip-all <ім'я виконуваного файлу>*.

Зауважимо, що всі ці особливості не відіграють істотної ролі в процесі вивчення основ ООП, але матимуть значення у разі створення нетривіальних проектів.

**Несумісність кодування.** Під час роботи в середовищі Lazarus для всіх компонентів організовано підтримку тільки кодування Unicode, а саме його різновиду UTF-8. Це означає, що всі введені користувачем рядкові дані матимуть таке саме кодування а в загальному випадку не будуть правильно опрацьовуватися стандартними підпрограмами обробки рядків, як-от: *Length*, *Pos*, *Copy* тощо (ці підпрограми будуть застосовні тільки для текстів латиницею).

Замість стандартних підпрограм потрібно використовувати підпрограми *UTF8Length*, *UTF8Pos*, *UTF8Copy*, *UTF8Delete*, *UTF8Insert*, *UTF8LowerCase*, *UTF8UpperCase*. Вони містяться в модулі *LCLProc* і мають аналогічні до стандартних (без префікса *UTF8*) підпрограм роботи з рядковими величинами призначення та параметри. Для звернення до деякого символу рядкової змінної з кодуванням UTF-8, наприклад для отримання *i*-го символу рядка *s*, замість програмного коду *s[i]* необхідно використовувати код *UTF8Copy(s,i,1)*.

Слід також пам'ятати, що в ОС Windows (до XP включно) використовується кодування CP1251 для багатьох рядкових ресурсів (імен файлів, повідомлень системи, більшості текстових файлів), тому для правильної обробки їх рядковими підпрограмами з префіксом *UTF8* рядкові ресурси необхідно перекодувати в формат UTF-8.

Для перекодування рядкових величин, що мають системне кодування (кодування, визначене стандартним в операційній системі), в кодування UTF-8, можна використовувати функцію *function UTF8Encode(const s : WideString):*

*UTF8String*, яка містяться в модулі *FileUtil*.

Для обробки файлів з кириличними іменами потрібно використовувати підпрограми *FileExistsUTF8*, *FileAgeUTF8*, *DirectoryExistsUTF8*, *ExpandFileNameUTF8*, *FindFirstUTF8*, *ExtractShortPathNameUTF8*, *FindNextUTF8*, *FindCloseUTF8*.

Щоб не витрачати обчислювальних ресурсів під час роботи програми на перекодування в тому разі, коли воно не потрібне (наприклад, якщо програма призначена для ОС Linux), можна скористатися перевіркою ОС:

```
{$IFDEF WINDOWS}
```

```
Оператор з перекодуванням
```

```
{$ENDIF}
```

Це означатиме, що код, пов'язаний з перекодуванням, буде включено до програми лише під час компіляції в ОС Windows.

**Додавання нових компонентів.** Щоб додати новий компонент до середовища Lazarus, потрібно перекомпілювати всю візуальну систему програмування, оскільки компонент тісно інтегрується в середовище, стає його частиною.

Таким чином, середовище Lazarus цілком придатне для вивчення ООП, а його безкоштовність та відкритість коду, можливість використання в комерційних розробках на законних засадах, портування власних додатків на найбільш популярні платформи дають змогу використовувати це середовище для створення досить складних кросплатформених проектів.

#### ЛІТЕРАТУРА

1. **Алексеев Е.** Free Pascal и Lazarus : Учебник по программированию / Е. Алексеев, О. Чеснокова, Т. Кучер – М. : ALT Linux; Издат. ДМК-прес, 2010. – 440 с.

2. **Горошко Ю.** Проблемы та особливості впровадження вільного програмного забезпечення у навчальний процес / Ю. Горошко, А. Костюченко, М. Шкардибарда // Комп'ютер в школі та сім'ї. – 2010. – № 7. – С. 8–10.

3. **Калверт Ч.** Руководство разработчика / Ч. Калверт, М. Калверт, Д. Кастер; [пер. с англ.] – М. : Вильямс, 2002. – 880 с.

#### ДО УВАГИ ОСВІТЯН, НАУКОВЦІВ, РЕКЛАМОДАВЦІВ!

Редакція журналу приймає замовлення на розміщення реклами, випуск тематичного номера або окремого розділу коштом замовника.