

Урок 21. Знайомство із середовищем програмування

Вивчення нового матеріалу.

<p>Слайд №1</p>	<p style="text-align: center;">Поняття мови програмування</p> <p>Задача → Алгоритм → Програма → Виконавець</p> <p>Запис мовою програмування</p> <p>Виконання програми</p> <p>Мова програмування складається зі знаків, слів і правил їх сполучення, як і звичайна мова</p> <p>Мовою програмування записують алгоритми</p> <p>Мова програмування – це мова для запису алгоритмів, що виконуватимуться комп'ютером</p>
<p>Слайд №2</p>	<p style="text-align: center;">Існують сотні мов програмування, як і звичайних мов. Ось найпопулярніші з них</p> <ul style="list-style-type: none">1969 • Pascal – одна з найпопулярніших у світі навчальних мов. Названа на честь французького вченого XVII століття Блеза Паскаля1972 • C (читається "Сі") – дає змогу гнучко обробляти дані (на рівні окремих бітів), керувати пам'яттю тощо1990 • Java – призначена насамперед для розробки веб-програм2002 • C# (читається "Сі шарп") – мова C із доданими засобами об'єктно-орієнтованого програмування2004 • Python – проста й водночас потужна мова. Часто використовується для навчання

Знайомство з середовищем візуального програмування LAZARUS

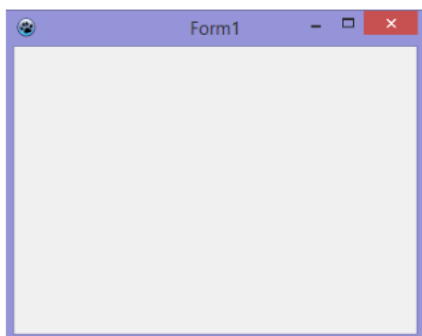
Середовище **LAZARUS** призначене для створення програм мовою Pascal. З його допомогою ви можете створити такі самі програми, якими щодня користуєтеся.



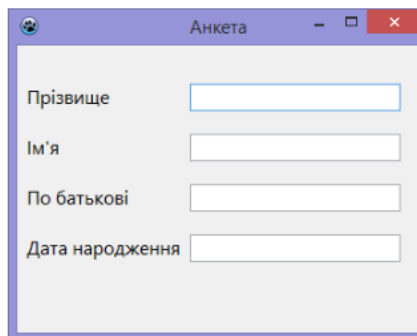
Звісно, для створення серйозних програм потрібні відповідні знання, однак створити прості програми може кожен з вас.

У Lazarus ми будемо створювати Windows-програми,
а кожна така програма виконується у вікні.

Найпростіше вікно – це вікно **форми**

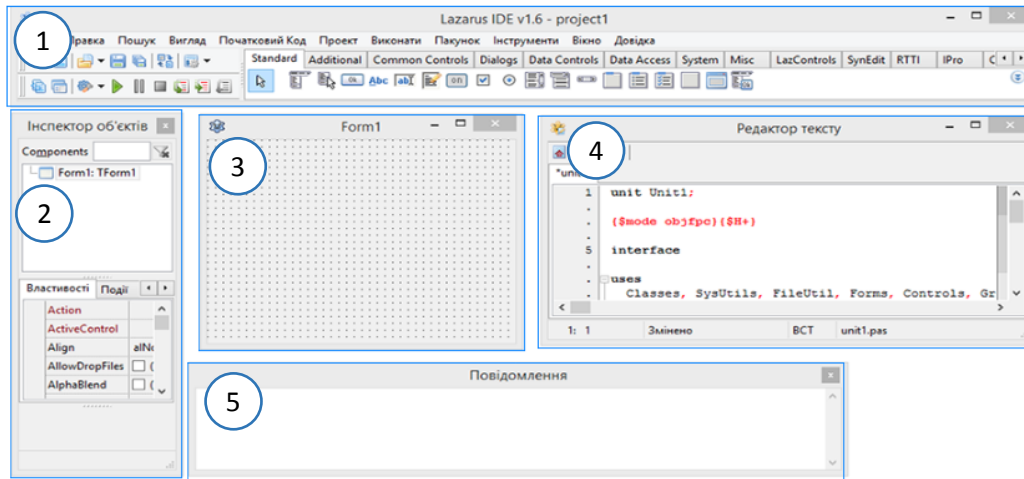


Вікно порожньої форми



Вікно форми з **елементами керування**

Після запуску програми відкриваються такі вікна:



1 Панель містить головне **меню** середовища, **панель інструментів** та компонентів. Саме тут розміщено всі інструменти для управління середовищем та створюваною програмою.

2 Вікно **Інспектор об'єктів** містить список усіх використаних об'єктів, а також перелік їх властивостей та можливих подій, на котрі може реагувати кожен із цих об'єктів.

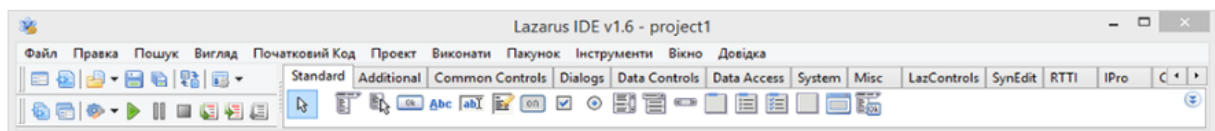
3 Вікно **Форма** – основний об'єкт програми. Саме на формі можна розміщувати всі ті елементи, котрі ви хочете бачити у вікні майбутньої програми (різноманітні кнопки, текстові поля, зображення, текстові написи тощо). Програма може мати кілька різних форм.

4 Вікно **Редактор тексту** призначене для написання тексту програми (**програмного коду**). Щоб спростити сприймання тексту, LAZARUS розфарбовує різні фрагменти коду різними кольорами. Деякі фрагменти коду програми LAZARUS створює автоматично.

5 У вікні **Повідомлення** виводитимуться повідомлення про хід виконання програми, зокрема повідомлення про помилки

Призначення вікон Lazarus

Усі вікна "плаваючі", їх можна вільно переміщувати у довільне місце екрана, згорнути чи закрити. Якщо закриту панель потрібно знову відкрити, це робиться за допомогою відповідних пунктів меню **Вигляд**.



Зауважте, що після першого запуску середовища Lazarus автоматично створюється новий **проект** (так називається створювана вами програма), якому надається ім'я **project1** – це ім'я відображається в заголовку середовища.

Слайд
№7

Якщо хтось уже працював на вашому комп'ютері в Lazarus, то після запуску середовища буде автоматично відкрито останній проект.

А для роботи з новим проектом його необхідно **створити**.

Слайд
№8

Створення нового проекту

- Відкрийте пункт меню **Проект**.
- Оберіть команду меню **Новий проект**.
- Клацніть кнопку **ОК**.
- У діалоговому вікні буде запропоновано зберегти попередній проект. Клацніть кнопку **Ні**. У результаті буде створено новий проект.

Слайд
№9

Збереження нового проекту

Розглянемо послідовність дій збереження нового проекту. Припустимо, що його необхідно зберегти в папці **8-A**.

- Відкрийте пункт меню **Файл**.
 - Оберіть команду **Зберегти як...**
 - У вікні, що відкриється клацніть двічі папку **8-A**.
 - Клацніть кнопку **Сохранить** для збереження файлу із розширенням **.LPI** (інформаційний файл проекту).
 - Клацніть кнопку **Сохранить** для збереження файлу із розширенням **.PAS** (програмний код мовою Pascal).
- У результаті проект буде збережено.

Слайд
№10

Після збереження в папці проекту буде створено кілька файлів, серед яких файли з такими розширеннями:

LPI (Lazarus Project Information) – інформаційний файл проекту;

PAS – програмний код мовою Pascal;

LFM – опис форми;

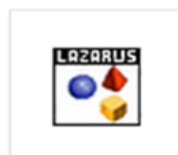
ICO – піктограма виконуваного файлу програми.



project1.ico



project1.lpi



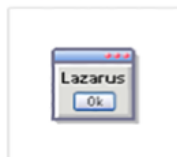
project1.lpr



project1.lps



project1.res



unit1.lfm



unit1.pas

Щоб наступного разу відкрити проект,
достатньо двічі клацнути файл із розширенням **LPI** – тоді автоматично відкриється
середовище LAZARUS, а в ньому – проект.



project1.ico



project1.lpi



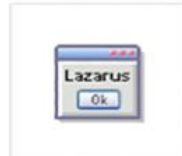
project1.lpr



project1.lps



project1.res



unit1.lfm



unit1.pas

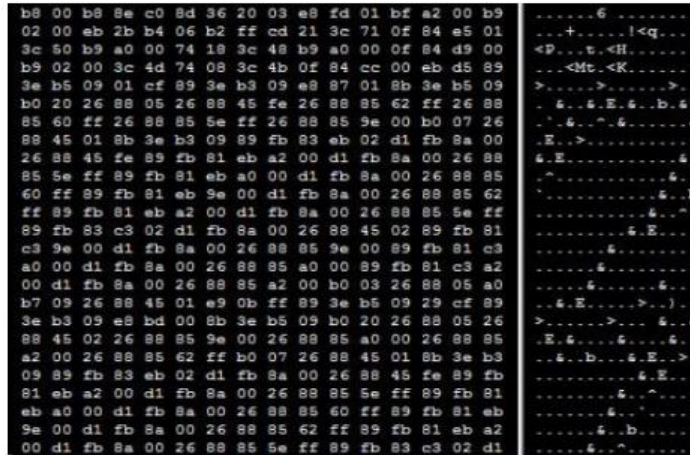
Вправа 1.

Вправа 1 у Lazarus

1. Відкрийте середовище Lazarus.
2. Створіть у ньому новий проект.
3. Збережіть проект.
4. Закрийте середовище, а потім відкрийте збережений проект.

Слайд №12

Насправді програміст пише програми зовсім не на тій мові, на якій їх здатен «зрозуміти» та виконати комп'ютер! Програми, які виконує комп'ютер, містяться у файлах з розширеннями **exe** або **dll** (такі файли ще називають **виконуваними**). Ось, подивіться, це – вміст типового exe-файлу:



Ви щось у цьому розумієте?
А комп'ютер розуміє саме такий «текст», його називають ще **машинним кодом**.

Слайд №13

Програміст створює та розуміє ось такий програмний код:

```
unit Unit1;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, FileUtil, Forms, Controls,

type

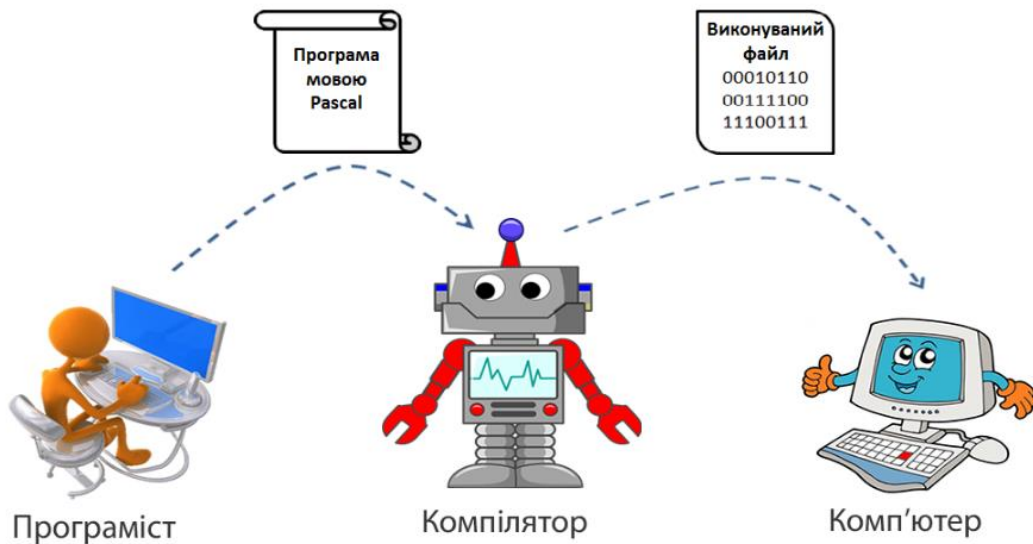
  { TForm1 }

  TForm1 = class(TForm)
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { private declarations }
  public
    { public declarations }
  end;
end;
```




Як же порозумітися програмісту і комп'ютеру?
Тут не обійтися без перекладача! Певна річ, автоматизованого.
Такий перекладач називається **компілятором**.

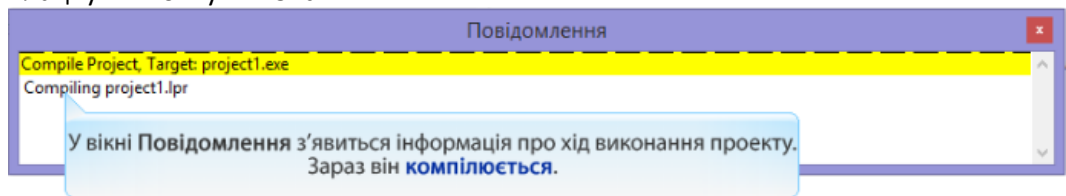
Ось так відбувається **компіляція** – перетворення тексту програми на виконуваний файл



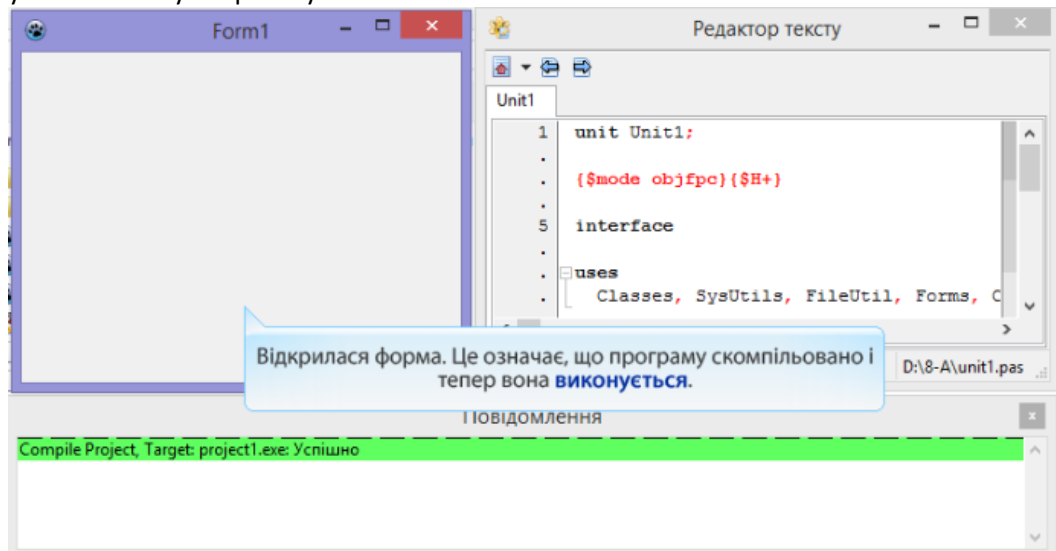
Компіляція програми

Щоб скомпільувати та виконати програму необхідно:

- Клацнути кнопку **Виконати** .

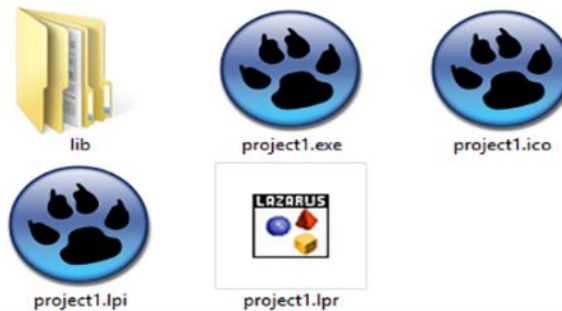


В результаті відкриється форма, а у вікні **Повідомлення** з'явиться інформація про успішний запуск проекту.



Форму запущеного проекту можна пересувати, згортати, змінювати розміри, закривати. Після закриття форми буде завершено роботу проекту. Це необхідно буде підтвердити натиснувши кнопку **Гаразд**.

Якщо тепер переглянути в папку проекту, побачимо в ній новий файл із розширенням **EXE**. Крім того, в папці з'явилася підпапка **LIB**.



EXE-файл – це виконуваний файл, який можна запустити без середовища LAZARUS.



А тепер перевірте, чи буде працювати програма, якщо перенести файл **project1.exe** в іншу папку і запустити його.

Вправа 2.

Вправа 2 у Lazarus

1. У середовищі Lazarus запустіть програму на виконання.
2. Перемістіть форму, згорніть та розгорніть її, змініть розмір форми, закрийте форму.
3. Перегляньте вміст папки проекту.

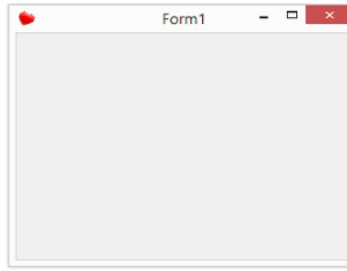
Новий матеріал до вправи 3.

Зміна піктограми

Розглянемо послідовність дій для зміни піктограми форми та exe-файлу програми.

1. Закачайте з Інтернету або знайдіть на комп'ютері будь-який файл із розширенням **.ICO** та розмістіть його в деякій папці на вашому комп'ютері.
2. Відкрийте меню **Проект**.
3. Виберіть пункт **Параметри проекту**.
4. У вікні **Параметри для проекту** оберіть вкладку **Параметри додатку** та клацніть кнопку **Завантажити значок**.
5. У вікні **Відкрити існуючий файл** двічі клацніть файл піктограми.
6. Натисніть **Открить**.
7. Так ви оберете новий значок для своєї програми. Підтвердіть вибір кнопкою **Гаразд**.
8. Щоб ці зміни відобразилися на формі, програму потрібно повторно запустити на виконання. Натисніть кнопку **Виконати**.

Після повторного запуску програми вибрана піктограма замінить стандартну в заголовку форми та **EXE**-файлі.



Слайд
№18

Зміна заголовку форми.

Розглянемо послідовність дій для зміни заголовку форми.

1. На панелі **Інспектор об'єктів** на вкладці **Властивості** оберіть властивість **Caption**.
2. Введіть напис, наприклад **Вправа 3** та натисніть **Enter**.
В результаті напис **Вправа 3** з'явиться в заголовку форми.

Вправа 3.

Вправа
№3

Вправа 3 у Lazarus





1. Змініть піктограму заголовку форми та exe-файлу.
2. Змініть заголовок форми.

Вивчення нового матеріалу.

Слайд
№19

Зауважте, що в папці проекту з'явилася нова підпапка **BACKUP**

У ній LAZARUS зберігає стару, попередню версію програми «про всяк випадок», якщо ви раптом захочете до неї повернутись

 project1.lpi.bak
 unit1.lfm.bak
 unit1.pas.bak
 project1.lps.bak

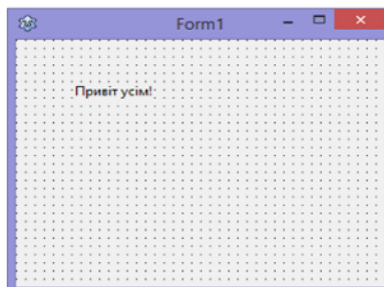


Щоб повернутися до збереженої попередньої версії програми, видаліть розширення **.BAK** у файлів з папки **BACKUP** і скопіюйте ці файли у папку з проектом.

Слайд
№20

Створення на формі напису

1. На панелі інструментів оберіть інструмент **Абс**.
2. Клацніть на формі
У результаті на формі з'явиться напис, текст якого можна в подальшому змінити.
3. Для цього на панелі **Інспектор об'єктів** на вкладці **Властивості** оберіть властивість **Caption**.
4. Замість попереднього тексту уведіть напис **Привіт усім!** та натисніть **Enter**.

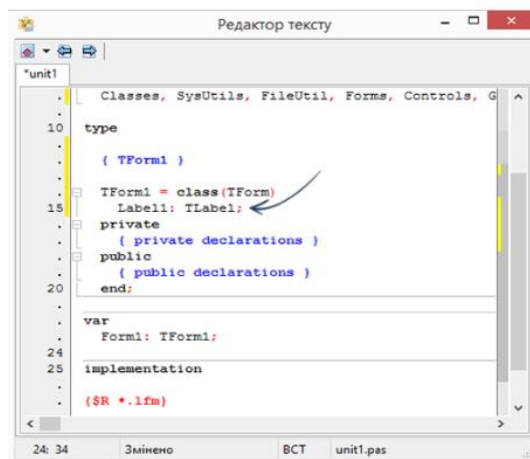


Отримаємо таку форму

Слайд
№21



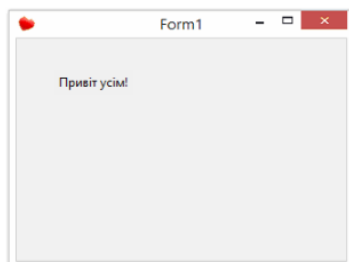
Зауважте, що на панелі Редактор тексту до програмного коду
додався рядок **Label1: TLabel;**



Слайд
№22



Після запуску програми на виконання форма
виглядатиме так:



Вправа 4.

**Вправа
№4**

Вправа 4 у Lazarus

Створіть на формі напис

Привіт усім!