

**Основні особливості  
середовища розробки.  
Візуальне конструювання.  
Структура проекту**



## Основні поняття

**Об'єктно-орієнтоване програмування** - це метод програмування, оснований на поданні програми у вигляді сукупності взаємодіючих об'єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії наслідування.

# ➔ Основні принципи ООП

- Все є об'єктами.
- Всі дії та розрахунки виконуються шляхом взаємодії (обміну даними) між об'єктами, при якій один об'єкт потребує, щоб інший об'єкт виконав деяку дію. Об'єкти взаємодіють, надсилаючи і отримуючи повідомлення. **Повідомлення** — це запит на виконання дії, доповнений набором аргументів, які можуть знадобитися при виконанні дії.
- Кожен об'єкт має незалежну пам'ять, яка складається з інших об'єктів.
- Кожен об'єкт є представником (екземпляром, примірником) класу, який виражає загальні властивості об'єктів.



# Середовище програмування Turbo Delphi

Середовище візуального програмування Delphi – це графічна автоматизована оболонка над об'єктно-орієнтованою версією мови Паскаль (Object Pascal). Якщо у мові Паскаль структурними одиницями є дані та команди, то тут такою структурною одиницею є візуальний об'єкт, який називається **компонентом**.

Автоматизація програмування досягається завдяки можливості переносити компонент на форму (у програму) з палітри компонентів і змінювати його властивості, не вносячи вручну змін до програмного коду.



# Середовище програмування Turbo Delphi

Це середовище розробки дає змогу використовувати велику **бібліотеку візуальних компонентів** (англ. *Visual Component Library (VCL)*). Це вже знайомі нам елементи керування: кнопки, поля, написи, прапорці, перемикачі, лічильники, списки, смуги прокручування, а також багато ін. Під час розміщення компонентів на формі автоматично генерується відповідний фрагмент програми мовою **Delphi**. Це значно зменшує і спрощує роботу зі створення програм. Такий спосіб розробки називають **візуальним**.

# ➔ Запуск Turbo Delphi

- Щоб відкрити **Turbo Delphi**, виконайте такі дії:
- Двічі клацніть по ярлику програми на Робочому столі;
- Запустіть файл TDrun.exe, знайшовши його за адресою: C:\Portable\TDLite\

# Елементи вікна



The image shows the Turbo Delphi Lite IDE interface with several components highlighted by callouts:

- Головне меню**: Points to the menu bar at the top (File, Edit, Search, View, Project, Run, Component, Tools, Window, Help).
- Швидкі кнопки**: Points to the toolbar below the menu bar.
- Structure**: Points to the Structure view on the left side of the IDE.
- Форма**: Points to the central design canvas (Form1).
- Project Manager**: Points to the Project Manager window on the right side.
- Object Inspector**: Points to the Object Inspector window at the bottom left, showing properties for Form1.
- Tool Palette**: Points to the Tool Palette window at the bottom right, showing a list of standard components.

The Object Inspector window shows the following properties for Form1:

Property	Value
<b>Action</b>	
Action	
Caption	Form1
Enabled	True
HelpContext	0
Hint	
Visible	False
<b>Drag, Drop and Docking</b>	
DockSite	False
DragKind	dkDrag
DragMode	dmManual
UseDockManager	False
<b>Help and Hints</b>	
HelpContext	0
HelpFile	
HelpKeyword	
HelpType	htContext
Hint	
ShowHint	False

The Tool Palette window shows the following components:

- Standard
  - Frames
  - TMainMenu
  - TPopupMenu
  - TLabel
  - TEdit
  - TMemo
  - TButton
  - TCheckBox
  - TRadioButton
  - TListBox
  - TComboBox
  - TScrollBar



# Середовище програмування Turbo Delphi

**Формою** називають компонент, який володіє властивостями вікна Windows і призначений для розташування інших компонентів. Компоненти на формі можуть бути *видимими* та *невидимими*. Перші призначені для організації діалогу з користувачем. Це різні кнопки, списки, текстові поля, зображення тощо. Вони відображаються на екрані під час виконання програми. Невидимі компоненти призначені для доступу до системних ресурсів комп'ютера.

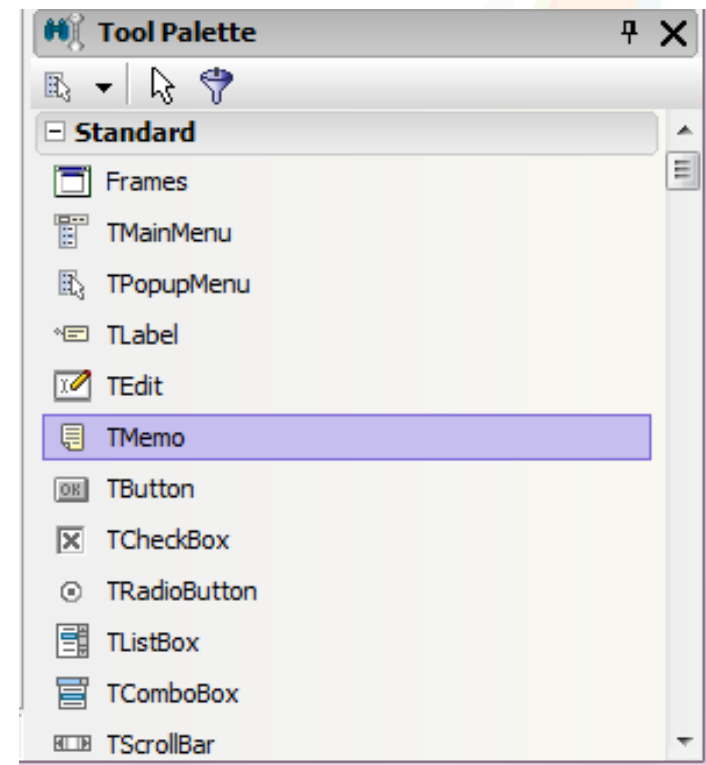




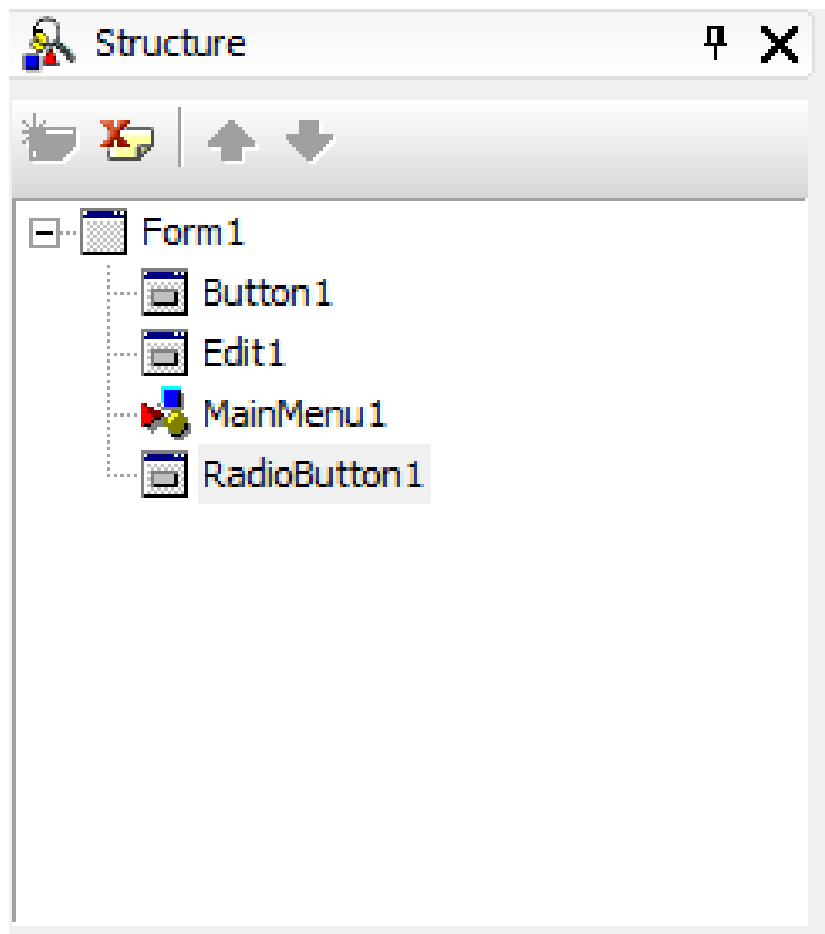
# Палітра інструментів Tool Palette

**Палітра інструментів** – це вікно, в якому відображається список об'єктів, доступних для використання в поточному режимі.

Щоб помістити об'єкт (компонент) у **центрі** вікна форми, **двічі** клацають на його піктограмі. Якщо потрібно розташувати компонент **десь на формі**, клацають **один раз** на його піктограмі і один раз у потрібному місці форми. Вибраний компонент можна переміщати на формі, а також змінювати розміри, перетягуючи його маркери.

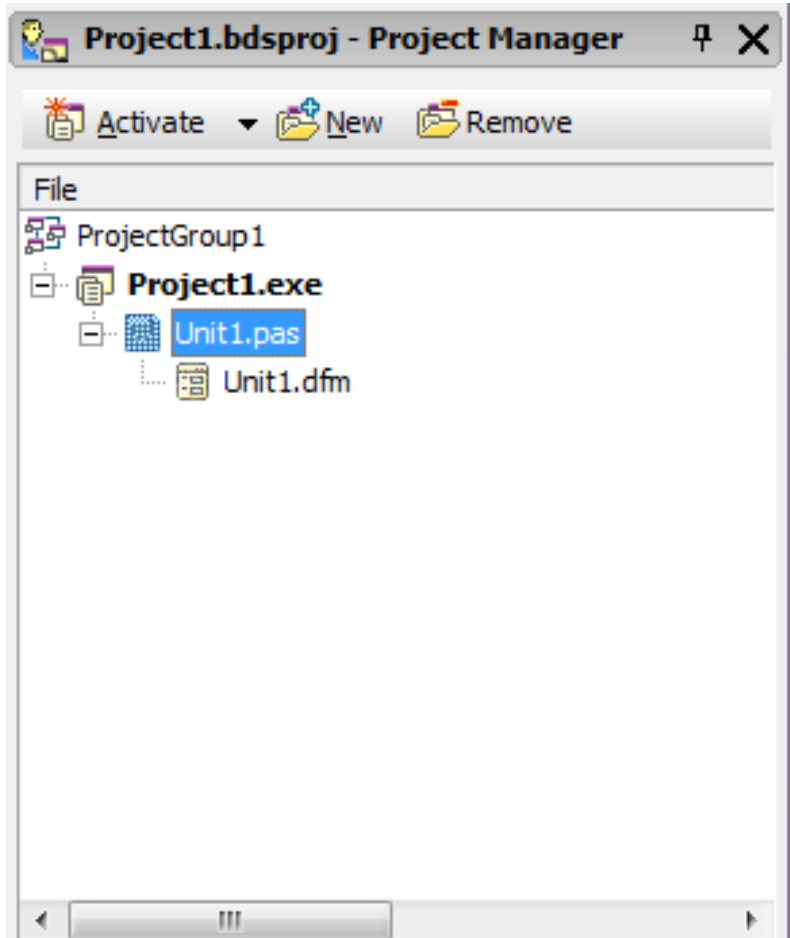


# ➔ Вікно об'єктів



**Вікно об'єктів (Structure) –**  
вікно з умовним зображенням усіх компонентів, поміщених програмістом у проект.

# ➔ Project Manager



**Project Manager**  
(англ. *project manager* –  
управляющий проектом) –  
у ньому  
відображається  
структура файлів  
проекту

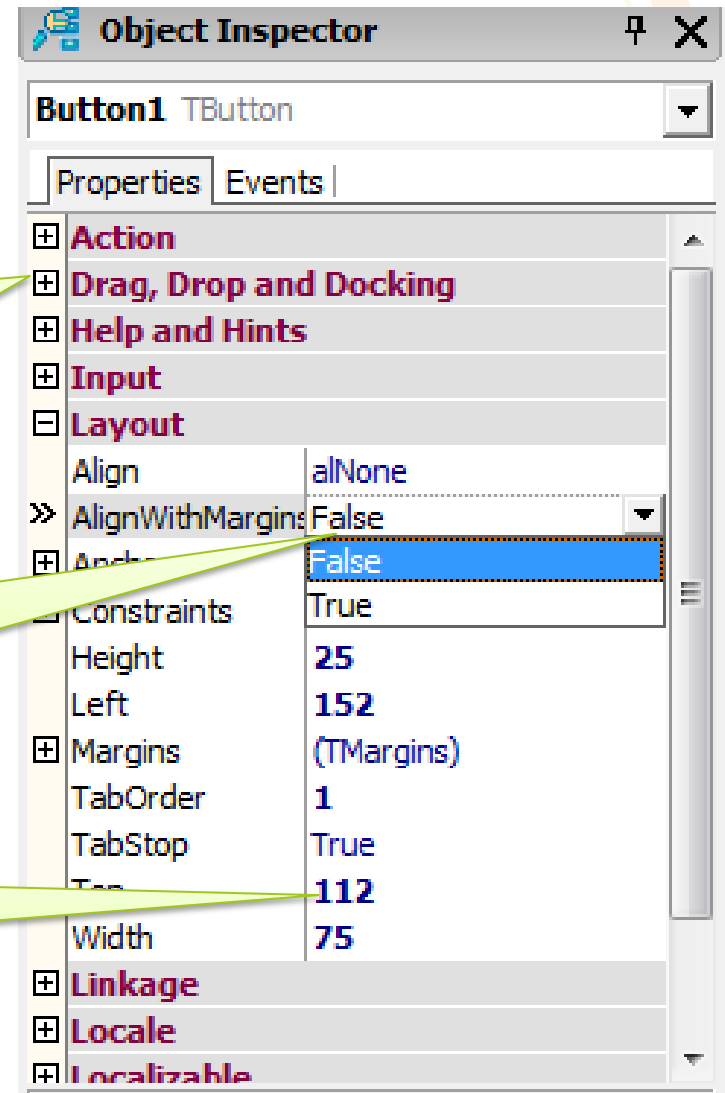
# ➔ Вікно Інспектора об'єктів

Складається з двох закладок: **Properties** і **Events**. У першій можна налаштувати властивості компонентів, у другій – писати процедури обробників подій.

*Вкладені властивості, що підтримують вкладені об'єкти.*

*Властивості, які вибирають значення зі списку.*

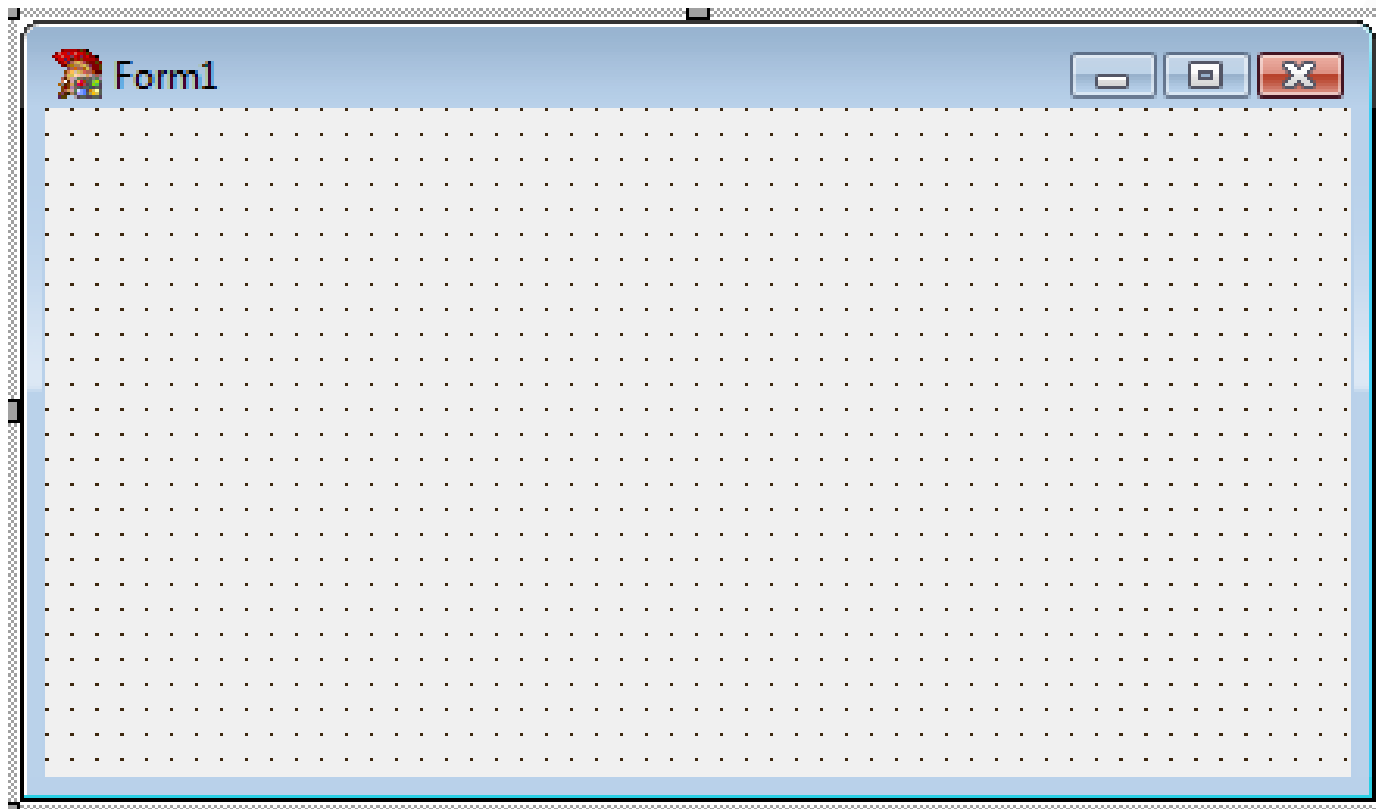
*Прості властивості, які є числами або рядками*



# Вікно форми



**Вікно форми** – редаговане зображення майбутнього інтерфейсу програми. Як ви в процесі проектування розташуєте компоненти на формі, так вони й будуть розташовані в запусненій на виконання програмі.





# Вікно редактора коду

Вікно, в якому міститься програмний код виконуваного завдання. Програміст пише цей код самостійно.

```
Unit1
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs;

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

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  ;
end;

end.
```

## ➤ Структура проекту

**Проектом** називають сукупність файлів, з яких Delphi створює готову для виконання програму. До складу кожного проекту обов'язково входять наступні файли:

- *файл проекту із розширенням\*.bdsproj*. Це невеликий файл з програмним кодом мовою **Object Pascal**, який містить посилання на всі файли проекту та ініціалізує програму;

## ➔ Структура проекту

- *файли опису всіх форм, які входять у проект:*  
файл модуля **\*.pas** і файл форми **\*.dfm**. Будь-якій формі проекту відповідає власний модуль;
- *файл ресурсів програми **\*.res**. У ньому описані ресурси, які не входять у форму, наприклад, піктограма програми;*



## ➔ Структура проекту

Для збереження **Delphi-проекту** необхідно задати імена модулів (автоматично пропонуються імена Unit1.pas, Unit2.pas, ...) та ім'я проекту (**Project1.bdsproj**). Ці імена можна змінити на власні. Для переміщення Delphi-проекту на інший комп'ютер необхідно мати лише файли таких типів: \*.dpr, \*.dfm, \*.pas, \*.res. Інші файли створюються автоматично.



Головний файл проекту є текстовим файлом, що містить **програмний код**, записаний мовою **Object Pascal**. Текстовий файл підключає всі використовувані програмні модулі й містить оператори для запуску додатка. Під час створення нового додатка Delphi автоматично генерує файл проекту. **Код файлу** проекту, що містить одну форму виглядає так:

```
program Project1;  
uses  
    Forms,  
    Unit1 in 'Unit1.pas' {Form1};  
{$R *.res}  
begin  
    Application.Initialize;  
    Application.CreateForm(TForm1, Form1);  
    Application.Run;  
end.
```

# ➔ Текст модуля формы

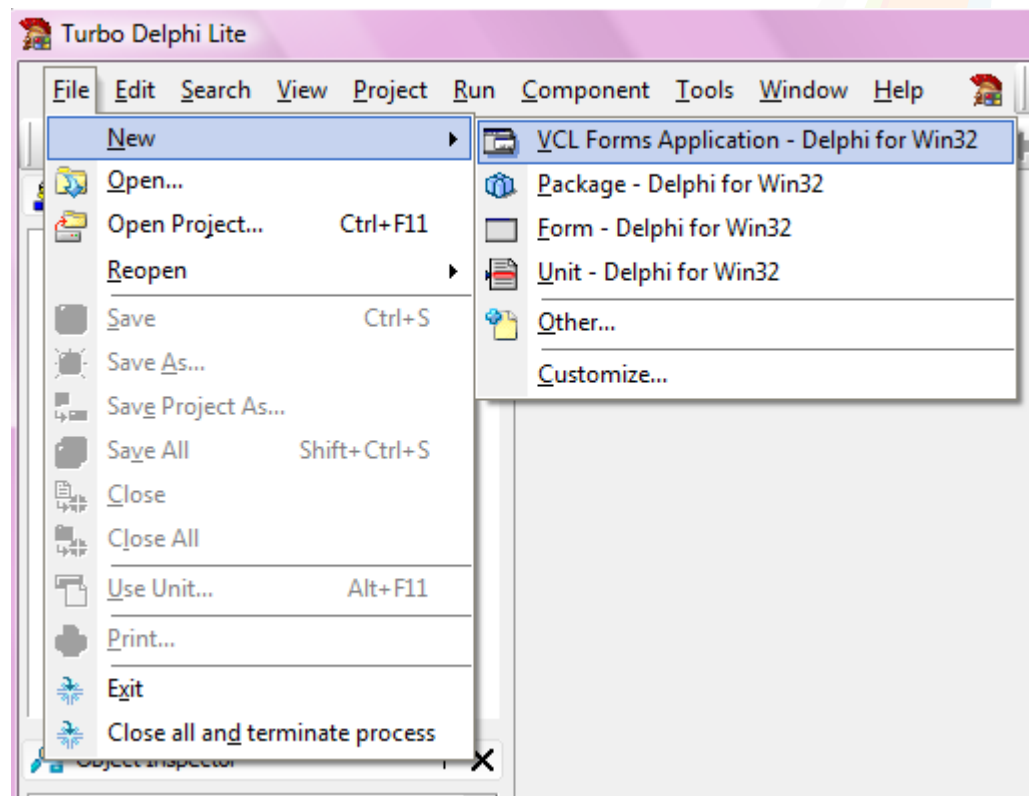
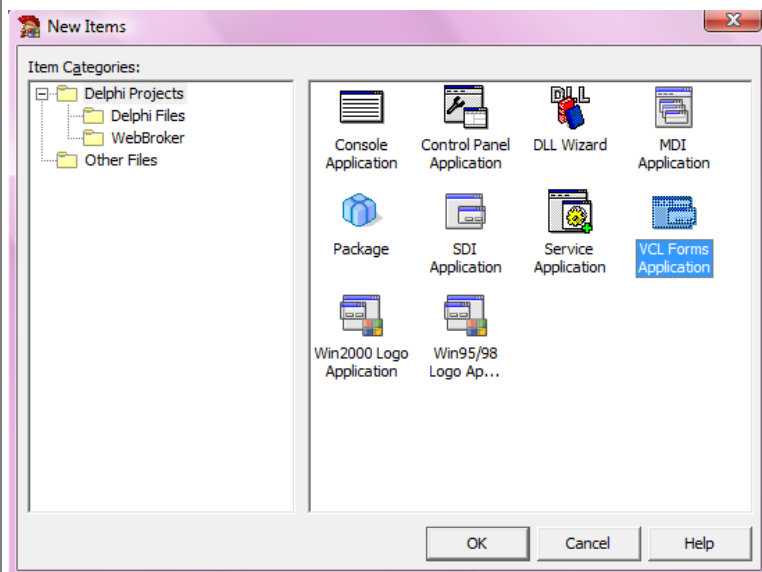
```
unit Unit1;  
interface  
uses  
    Windows, Messages, SysUtils, Variants, Classes,  
    Graphics, Controls, Forms, Dialogs;  
type  
    TForm1 = class(TForm)  
    private  
        { Private declarations }  
    public  
        { Public declarations }  
    end;  
var  
    Form1: TForm1;  
implementation  
{$R *.dfm}  
end.
```

# Створення найпростішого проекту

Для створення нового проекту слід виконати команду: **File ⇒ New ⇒ VCL Forms Application.**

або натиснути 

та у вікні **New Items** обрати **VCL Forms Application**





# Компіляція та збирання проекту

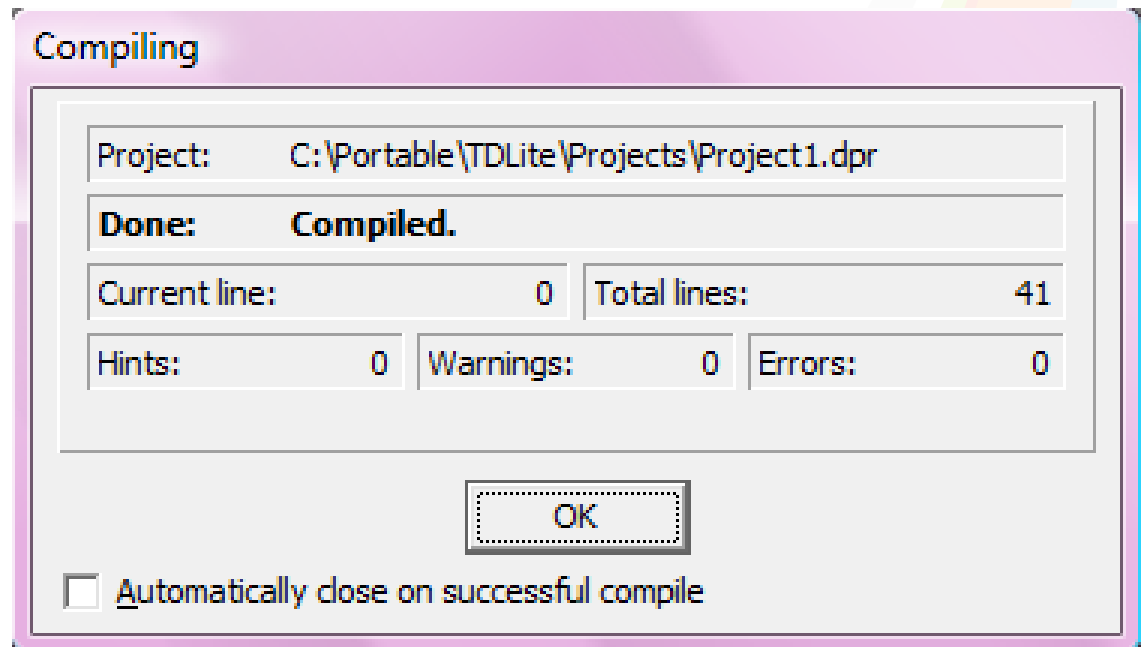
Компіляцію та збирання проекту можна виконати на будь-якій стадії його розробки.

**Компіляцією** називають одержання об'єктних модулів з вихідних текстів програмних модулів.

**Збирання** передбачає одержання виконуваного файлу з об'єктних модулів.

У середовищі **Turbo Delphi** компіляція і збирання проекту поєднані.

Для виконання  
компіляції  
використовують  
команду **Project**  
⇒ **Compile ...**  
або комбінацію  
клавiш **Ctrl+F9**.





➔ **Побудова застосунку**

Виконайте команду **Project** ⇒  
**Build Project1** (Побудувати  
Project1)

# ➔ Успішна побудова

Compiling

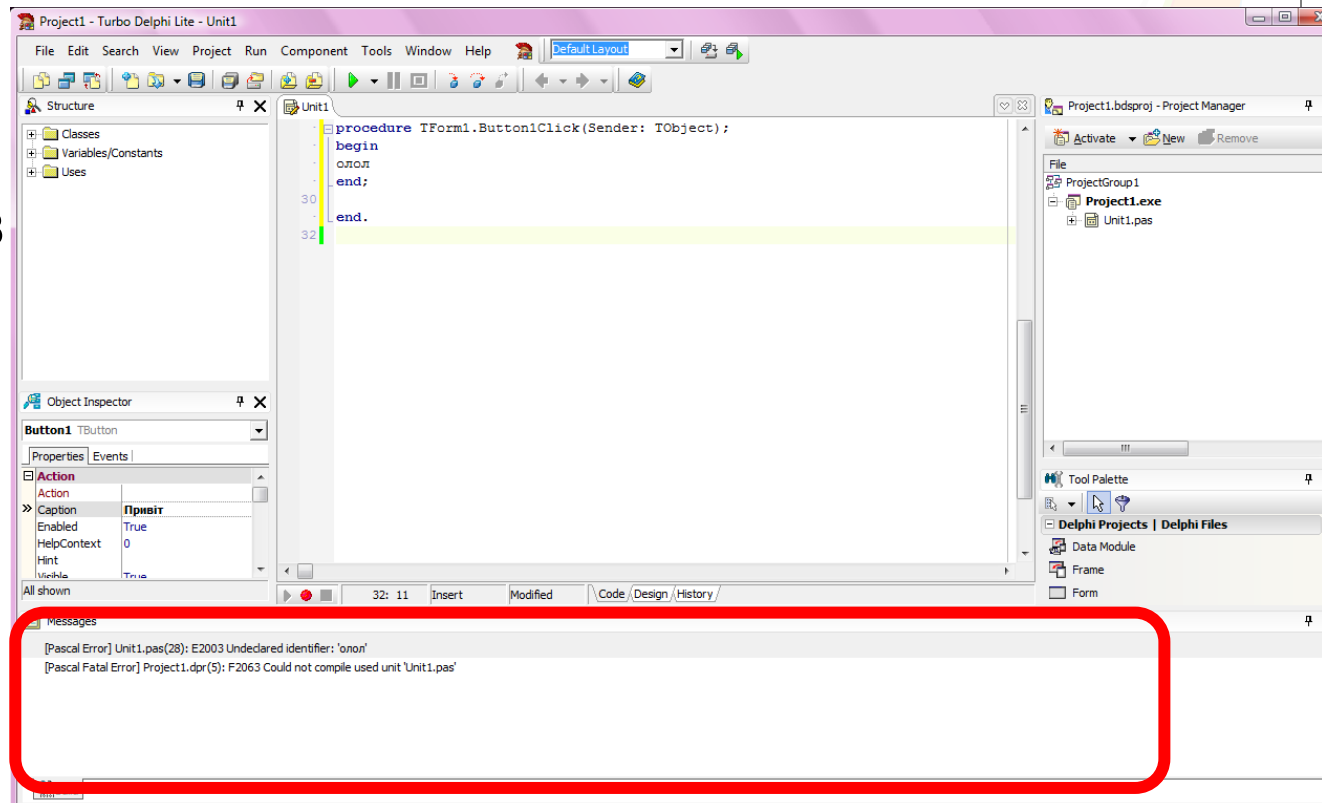
Project:	C:\Portable\TDLite\Projects\Project1.dpr		
Done:	<b>Build All</b>		
Current line:	0	Total lines:	41
Hints:	0	Warnings:	0
		Errors:	0

OK

Automatically close on successful compile

# ➔ Невдала побудова

Якщо при написанні коду допущена помилка, Delphi виділить рядок з помилкою червоним кольором. А зміст помилки буде відображено у нижній частині вікна.





## Запуск програми

- На панелі головного меню виберіть команду **Run (Виконання)**.
- Виконайте команду **Run (Виконати)** або натисніть клавішу **F9**.

## ➔ Збереження проекту

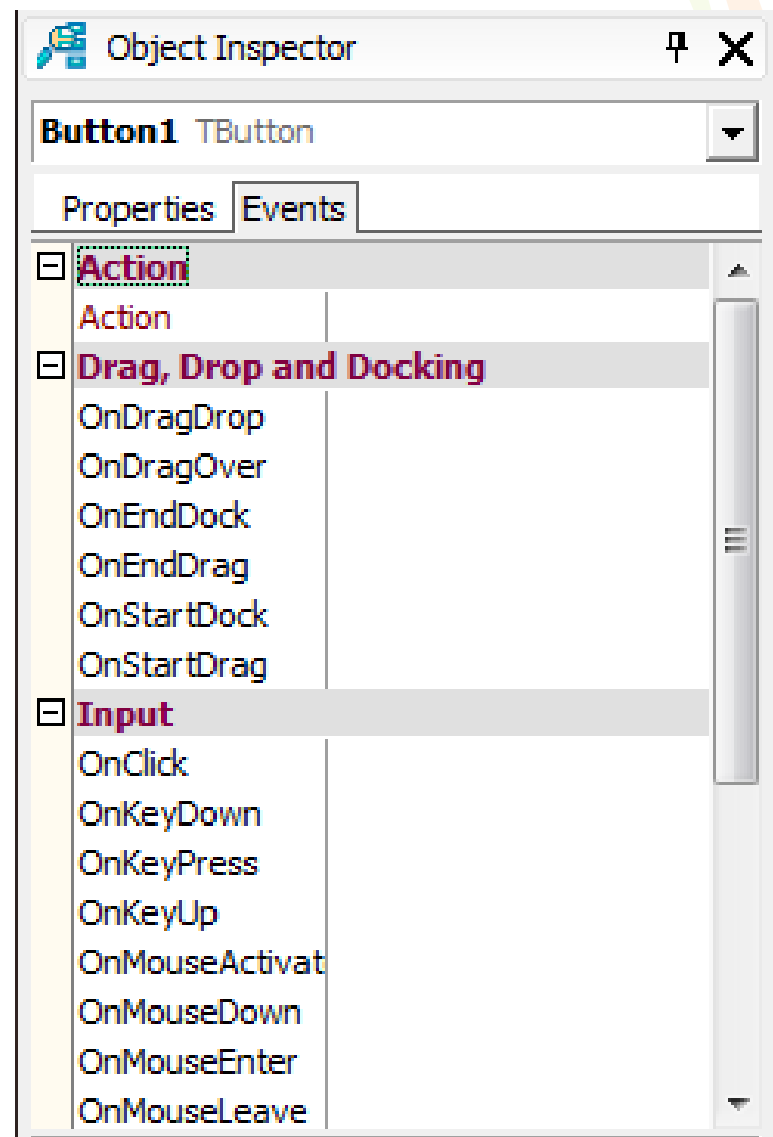
- У головному меню виберіть пункт **File** (Файл).
- Виконайте команду **Save All** (Зберегти все).
- Для виходу із середовища, виберіть меню **File** (Файл), а в ньому – команду **Exit** (Вийти).

# ➔ Події

Події активізуються діями користувачів. Наприклад, коли користувач клацає кнопку, вибирає елемент у полі зі списком або змінює текст у полі, відбувається **подія**.

Вона також відбувається, коли користувач клацає один раз або двічі елемент керування чи наводить на нього або знімає з нього покажчик миші.

Виникнення кожної з цих подій приводить до виконання зв'язаного з нею Delphi-коду, який називають **обробником події**. Він «змушує» програму виконувати певні дії.

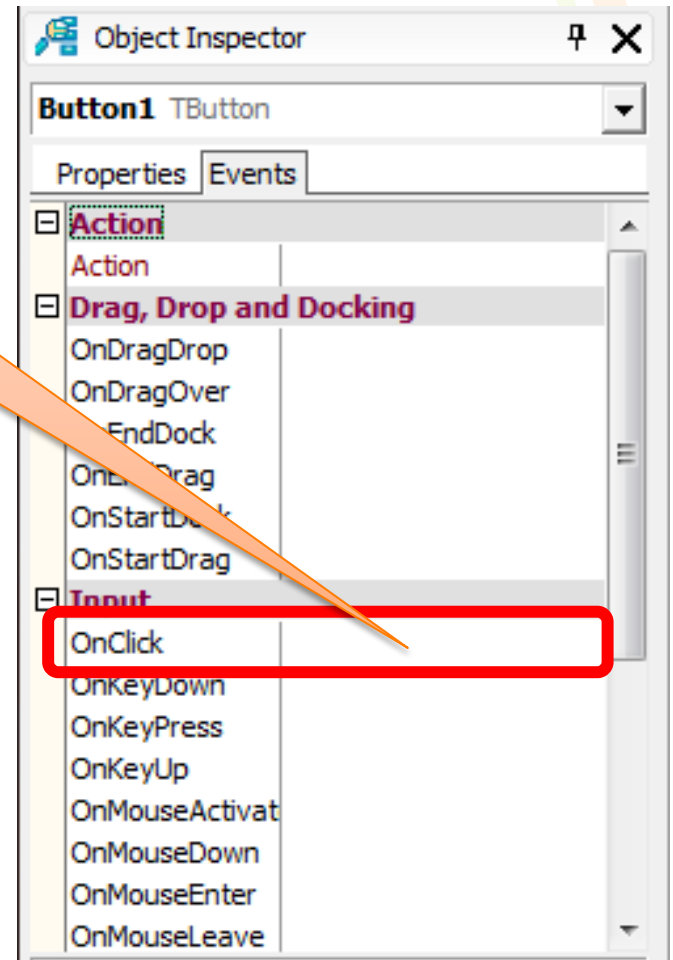


# ➔ Події

Двічі клацнути ЛКМ

```
Unit2  
-  
- procedure TForm2.Button1Click(Sender: TObject);  
- begin  
28 |  
- end;  
30  
- end.  
-
```

**Процедура** (лат. *procedere* – просуватися, йти вперед) – це частина програми, яка має ім'я та яку можна за цим іменем викликати на виконання в різних частинах програми.



## Домашнє завдання

- Опрацювати конспект.
- Встановити середовище візуального програмування на домашніх комп'ютерах
- Підготувати повідомлення «Історія мов програмування»