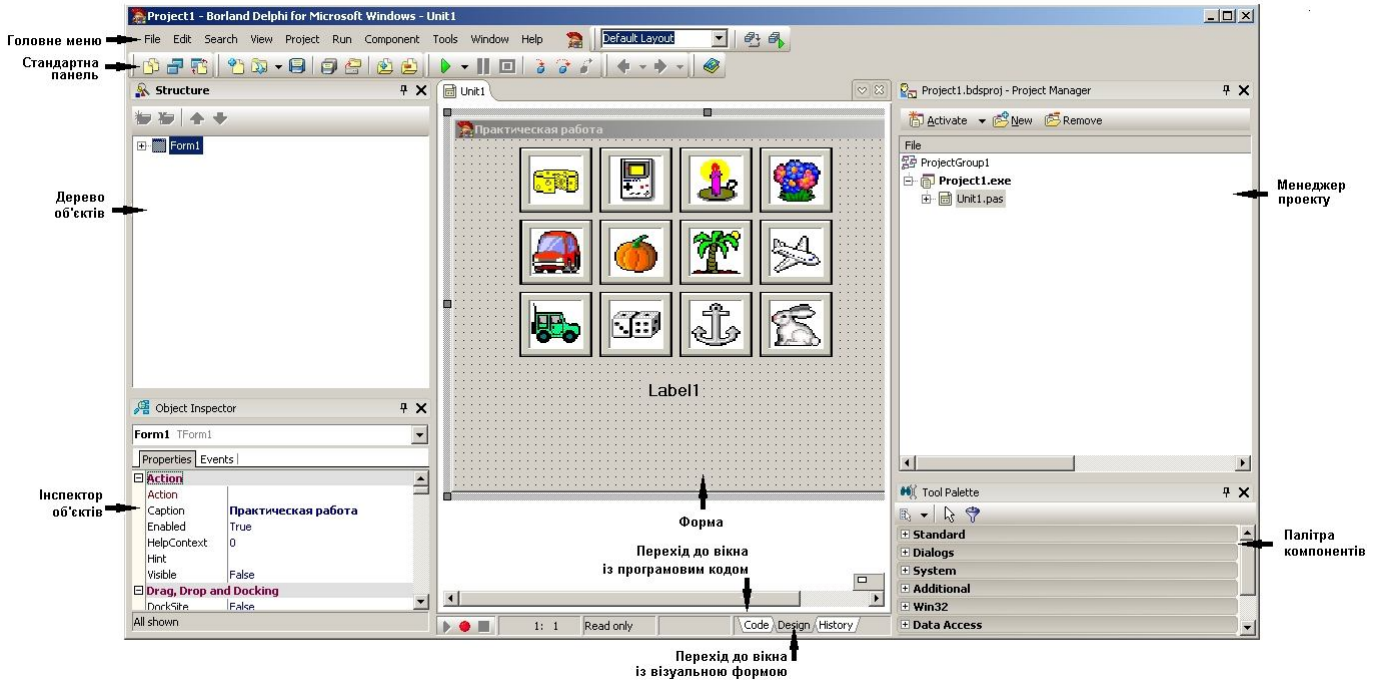
Тема: Створення простого проекту.

Мета: Познайомитися із візуальним середовищем програмування Delphi, створити власну форму, познайомитися із подіями, на які вона реагує, розробити просту процедуру для обробки подій; навчитися зберігати проекти на диску, розглянути, які файли містить проект, та вияснити їхнє призначення.

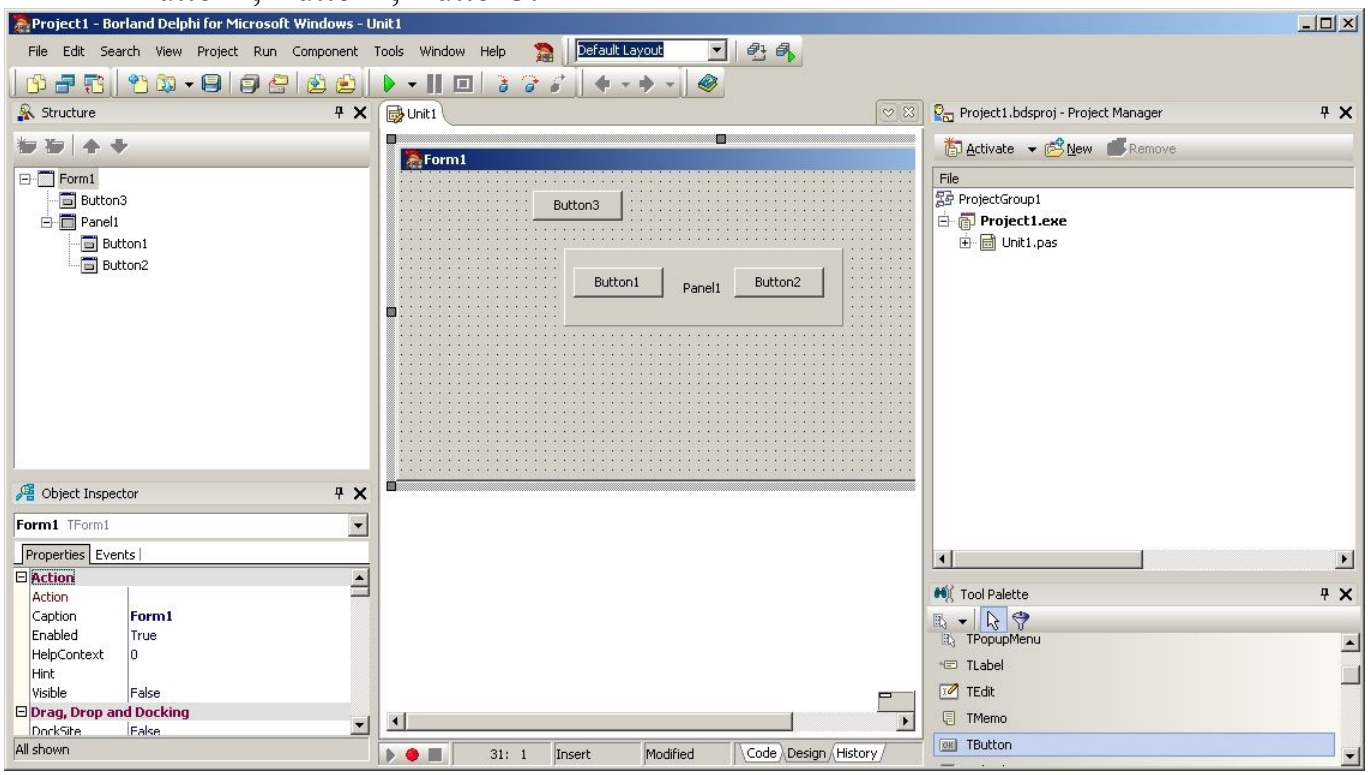
1. Знайомство із візуальним середовищем програмування. Елементи вікна середовища програмування.

Після успішного запуску середовища візуального програмування Delphi у робочому вікні можна виділити наступні елементи:

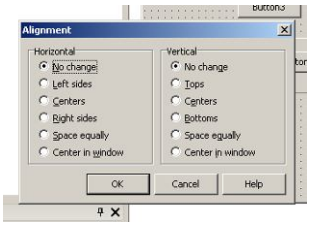


– У Головному меню зібрані основні керуючі пункти для редагування та запуску програми. Тут наявні уже знайомі по роботі із іншими додатками групи кнопок Завантажити (Open), Зберегти (Save), Редагувати (Edit), так і специфічні, які властиві лише мовним середовищам, наприклад, Проект (Project), Запуск (Run), які визначають режим компіляції та виконання програми.
– Нижче Головного меню розміщена Стандартна панель, у якій містяться швидку кнопки для виконання операцій, які найбільш часто зустрічаються при роботі із проектом.
– Палітра компонентів – це багатосторінковий діалоговий елемент, на кожній закладці якого розміщено декілька відеокомпонентів, які можна переносити на форму. Інтерфейс додатку визначається вибором ти чи інших компонентів. При проектуванні інтерфейсу додатку діє принцип WYSIWYG (What You See Is What You Get) – «що бачите, те й отримуєте». Оскільки компонентів велика кількість, то вони розміщені у Палітрі по закладках – Standard, Dialogs, System… Для того, щоб помістити компонент на форму, необхідно перейти на потрібну сторінку, клацнути мишкою на вибраному компоненті в Палітрі, а потім – у потрібному місці форми. Наприклад, нам

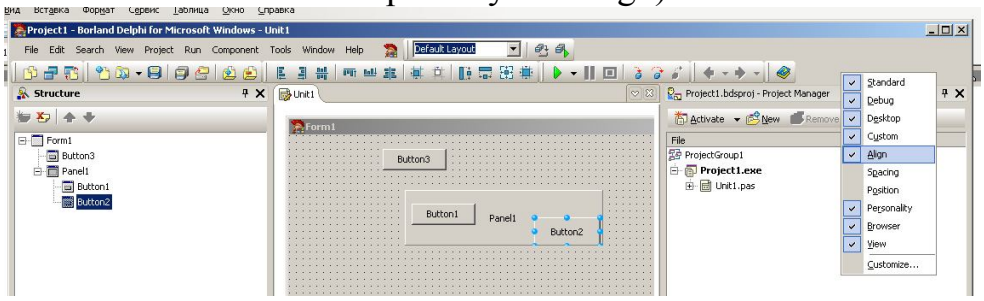
необхідно розмістити на формі три керуючі кнопки. Для цього із закладки Standard ми, переміщуючи іконку, переносимо на форму три компоненти; при цьому середовище програмування саме пронумерує ці компоненти: Button1, Button2, Button3.



Після перенесення компонента на форму можна змінювати місце його розміщення та розміри. За замовчуванням компоненти вирівнюються на формі по лініях сітки, крок сітки рівний 8 пікселам, і сітку видно на етапі проектування. Можливо підігнати місце розміщення компоненту із точністю до 1 піксела за допомогою клавіш-стрілок: при натиснутій клавіші Ctrl відбувається зміщення компоненту, а при натиснутій клавіші Shift – зміна його розмірів. Для деяких компонентів передбачено визначення місця їх розміщення на передньому або задньому плані. Вирівнювати розміщені компоненти можна або вручну, або виділити групу компонентів та скористатись командою Edit - Allign



Можна також скористатись активацією панелі Вирівнювання (викликати контекстне меню та вибрати пункт Align):

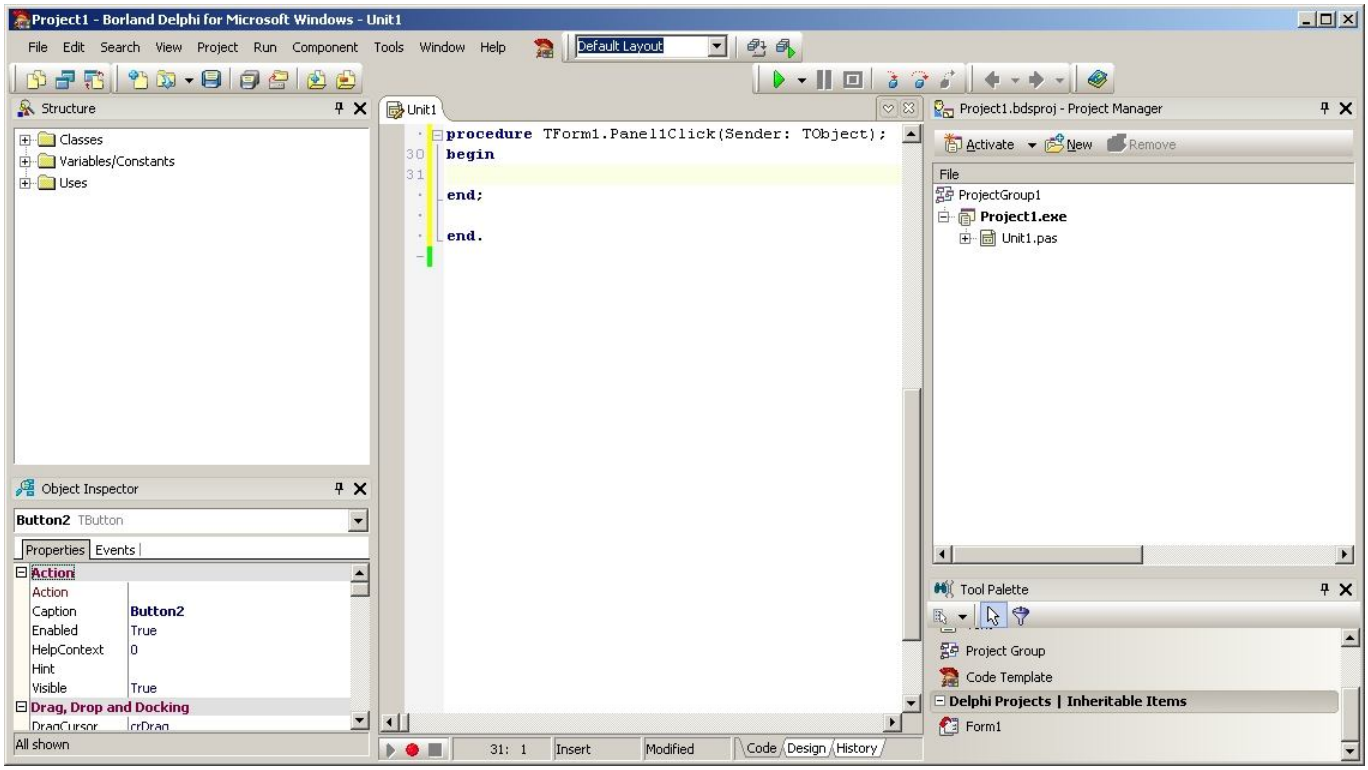


– Дерево об’єктів – вікно із умовними позначеннями всіх компонентів, розміщених програмістом у проекті. Це не просто перелік – якщо на формі є так звані об’єкти-контейнери (тобто, компоненти, до складу яких входять

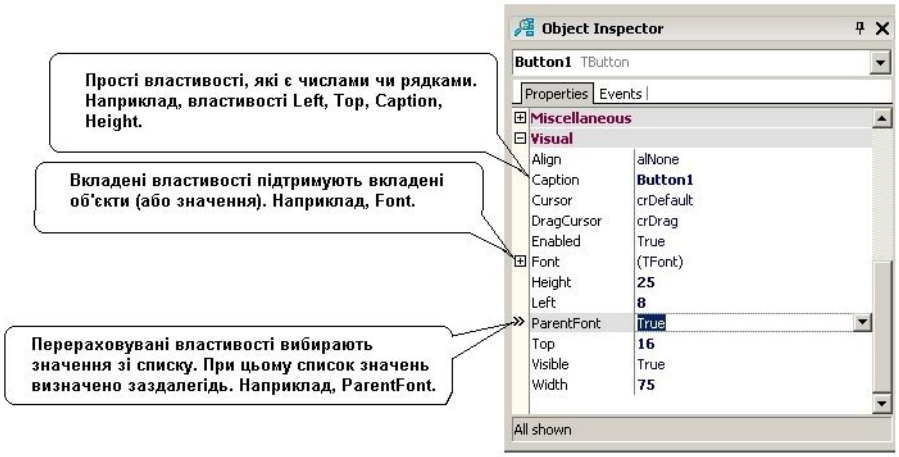
інші компоненти – панелі, сторінки, закладки тощо), то в даному вікні лініями зображені рівні їх підпорядкування.

Це важливо, наприклад, у тому випадку, коли Ви зробите невидимим деякий об'єкт-контейнер – тоді всі підпорядковані йому компоненти також стануть невидимими. Таким чином, Ви винесли на форму кнопку. Подивившись у вікно дерева об'єктів, легко визначити, чи належить вона формі, чи деякому іншому об'єкту-контейнеру. Окрім того, із використанням дерева об'єктів досить зручно переміщувати компоненти із одного контейнера в інший за допомогою мишки.

– Основне місце на екрані займає вікно Форми – це редаговане зображення майбутнього інтерфейсу програми. Як Ви у процесі проектування розмістите відео компоненти на формі, так (у більшості випадків) вони будуть розміщуватись і в запущеній на виконання програмі.
– За допомогою перемикача Code викликається вікно редактора коду, в якому розміщується програмний код нашої задачі. Його Ви будете вносити самостійно. Візуальне програмування лише додає рядки опису в програму по мірі виставлення компонентів на форму. Проте середовище Delphi, безумовно, не може здогадатись, які дії Ви маєте бажання виконати при натискуванні тієї чи іншої кнопки, винесеної на форму, що Ви хочете від інших компонентів.



Кожен винесений на форму компонент володіє множиною важливих та корисних властивостей, значення яких визначені за замовчуванням, проте можуть бути налаштовані програмістом у відповідності із конкретною програмою. Список доступних властивостей компонентів міститься у вікні Інспектора об'єктів, яке складається із двох вкладок – Properties (Властивості) та Events (Події).



Важливо пам'ятати, що деякі властивості не можна змінювати на етапі – вони доступні лише під час виконання програми. Такі властивості можна змінювати лише програмовим шляхом. Властивості, які перераховані в Інспекторі об'єктів, дозволяється змінювати як на етапі проектування, так і під час роботи додатку.

2. Програмна розробка та файли, що входять до її складу.

У середовищі Delphi розроблюваний проект являє собою набір файлів, з яких створюється додаток. До складу будь-якого проекту входять мінімум 6 файлів:

- proect1.dpr – головний файл проекту, він формується системою при створенні нового додатку;
- unit1.pas – перший модуль (unit) програми, який автоматично з'являється на початку роботи;
- unit1.dfm – файл опису форми, він використовується для збереження інформації про перелік відеокомпонентів та їх розміщення на формі;
- proect1.res – файл ресурсів, у ньому зберігаються іконки, растрові зображення, курсори;
- proect1.dof – текстовий файл опцій для збереження установок, пов'язаних із даним проектом (наприклад, директив компілятора);
- proect1.cfg – файл конфігурації, він містить інформацію про стан середовища.

Окрім перерахованих, до проекту можуть відноситись файли із картинками, відеофрагментами, звуками, файли довідкової системи тощо. Якщо зберігати проект під іншим іменем, то, окрім файлу проекту, змінять назву й файли із розширеннями res, dof, cfg. Якщо змінити ім'я файлу модуля (pas), то зміниться й ім'я файлу опису форми (dfm). Добрим стилем програмування вважається використання імен, які несуть смислове навантаження.

Після компіляції програми у робочій папці додатково з'являться файли із розширеннями:

- dcu – відкомпільовані модулі;
- exe – виконуваний файл.

Головний файл проекту є текстовим файлом, який містить програмний код, записаний на мові Object Pascal. Текстовий файл підключає всі використовувані програмні модулі та містить оператори для запуску додатка. При створенні нового

додатку Delphi автоматично створює файл проекту. Код файлу проекту, який містить лише одну форму, наведено нижче:

```
Program Proect1;
Uses
        Forms,
        Unit1 in 'Unit1.pas' {Form1};
{$R *.res}
Begin
        Application.Initialize;
        Application.CreateForm (TForm1, Form1);
        Application.Run;
End.
```

У розділі Uses підключається системний модуль Forms та модуль форми Unit1. Назва форми наводиться в круглих дужках. Директива компілятора {$R *.res} здійснює підключення ресурсів до результуючого exe-файлу.

Тіло програми містить оператори, які готують додаток до роботи (ініціалізують), створюють форму та приступають до виконання додатку.

У ході створення нових форм вміст цього файлу змінюється автоматично. Вручну він корегується лише у виняткових випадках.

При створенні додатку Delphi генерує порожню форму, текст модуля якої наведено нижче:

```
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.
```

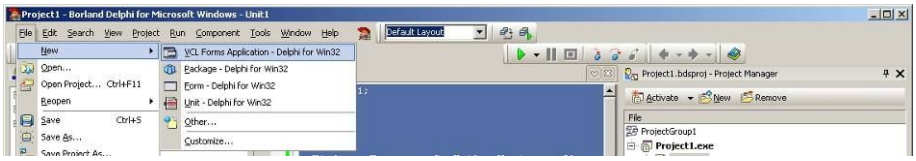
Модуль розпочинається із зарезервованого слова Unit, після якого записується ім'я модуля. Воно співпадає із іменем файлу, в якому зберігається модуль. У загальному випадку структура модуля має наступний вигляд:

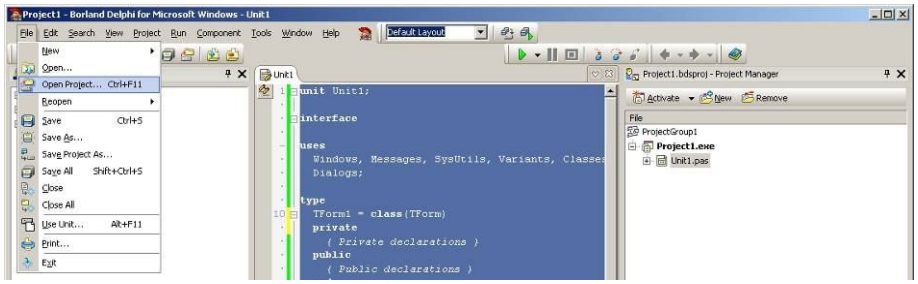
- заголовок;
- секція інтерфейсних оголошень interface;
- секція реалізації implementation;
- секція ініціалізації initialization;
- секція завершення finalization;
- ознака завершення тексту програми end.

3. Створення проекту, його компіляція, збереження, виконання.

При запуску Delphi автоматично створюється новий проект. При завантаженні іншого проекту (нового або того, щоб створений раніше) відкритий проект закривається.

Для створення нового проекту потрібно виконати команду File – New – VCL Forms Application – Delphi for Windows 32

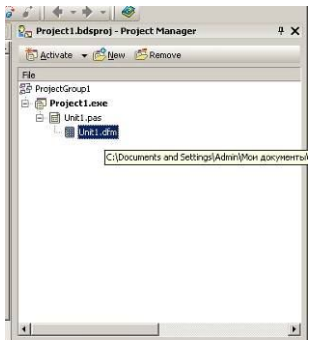




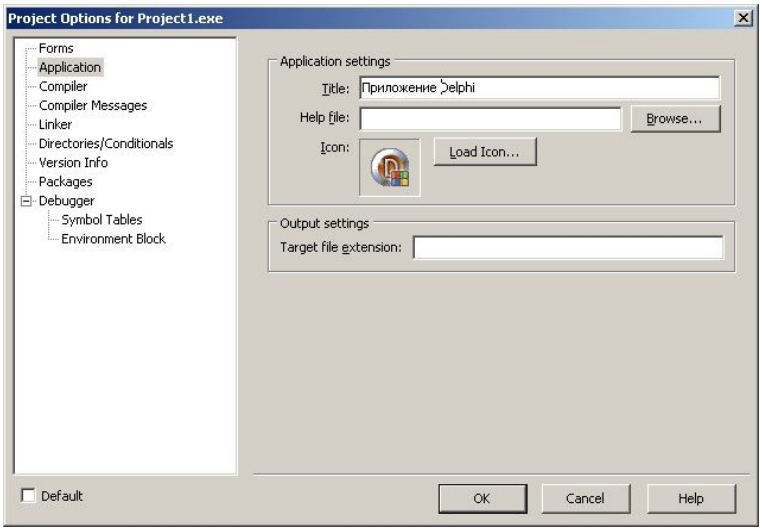
Для завантаження існуючого проекту потрібно виконати команду File – Open Project…

Рекомендується, створивши новий проект, відразу ж зберегти його в окремій папці.

Менеджер проекту Project Manager призначений для управління проектами та складовими частинами додатку, який Ви розробляєте. Менеджер проекту дозволяє працювати з групою проектів: можна переглядати, додавати чи вилучати проекти та їх складові частини. При додаванні (New) чи вилученні (Remove) проектів чи їх складових частин автоматично вносяться зміни й до відповідних файлів.



Встановити параметри проекту можна у вікні Project Options, яке відкривається за допомогою команди Project – Options:



На сторінці Forms можна назначити головну форму додатку та вибрати у списку Auto-create forms форми, які будуть створюватись одночасно із головною. Головна форма відкривається першою; її закриття призводить до закриття всього додатку.

На сторінці Application вказується назва та іконка, які будуть відображуватися у середовищі Windows на панелі задач, а також файл довідки, який підключається до проекту.

На сторінках Compiler та Linker встановлюються директиви компілятора та компоновщика (редактора зв'язків).

Компіляція та збірка проекту можуть виконуватися на будь-якому етапі його розробки. Під компіляцією розуміють отримання об'єктних модулів (dcu-файлів) із вихідних текстів програмних модулів (pas-файлів). Збірка полягає в отриманні виконуваного файлу із об'єктних модулів.

У середовищі Delphi компіляція та збірка проекту суміщені. Для виконання компіляції достатньо натиснути комбінацію клавіш Ctrl+F9. При цьому компілюються усі вихідні модулі, вміст яких змінювався після останньої компіляції. У результаті для кожного модуля створюється файл із розширенням dcu (Delphi Compiled Unit). Надалі середовище Delphi компілює головний файл проекту та компонує із dcu-файлів виконуваний файл (exe-файл), ім'я якого співпадає із іменем проекту.

4. Додавання декількох рядків коду до обробника події, їх аналіз.

Найбільш розповсюджений спосіб повідомити додатку про виникнення події – це клацнути мишкою під час роботи на будь-якому компоненті. При клацанні мишкою у програмі, яка працює, виникає подія OnClick. Якщо вона не опрацьовується програмою, клацання ні до чого не призводить. Щоб примусити програму реагувати на клацання, необхідно написати на мові Object Pascal фрагмент програми, який називається обробником події. При цьому створений фрагмент повинен являти собою послідовність команд, в яких програміст вказує, що саме повинна зробити програма у відповідь на клацання мишкою.

Щоб примусити Delphi самостійно зробити заготовку для процедури обробника події OnClick, необхідно у момент проектування додатку двічі клацнути на вставленому компоненті. Наприклад, для кнопки Button1 (компонент класу TButton) Delphi активізує вікно коду, в якому можна побачити наступний фрагмент:

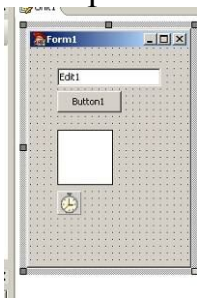
```
Procedure TForm1.Button1Click (Sender: TObject);
Begin


End;
```

Слово procedure повідомляє компілятор про початок методу класу, який реалізується у вигляді процедури. За ним слідує складене ім'я, яке містить ім'я класу TForm1 та власне ім'я методу Button1Click. При цьому у верхній частині модуля, всередині опису класу(розділ interface), з'явиться рядок:

Procedure Button1Click (Sender: TObject);

Виконані вказані дії призвели до виклику обробника OnClick – цей обробник найбільш часто використовується за замовчуванням (він викликається для більшості компонентів). Існують також компоненти, для яких за замовчуванням визначені інші обробники. Найбільш часто для форми використовують обробник OnCreate. Всередині цього програмного фрагменту зручно задавати які-небудь початкові значення для змінних, ініціалізувати параметри, відкривати файли та задавати

розміри компоненти, включаючи й саму форму. Компонент Edit, за допомогою якого здійснюється введення текстової інформації, за замовчуванням налаштований на обробник OnChange.

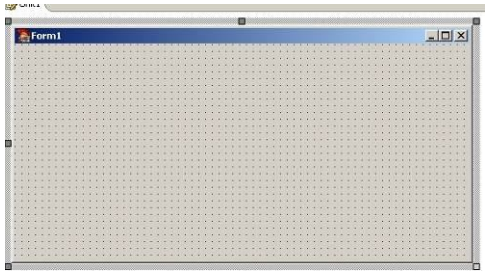



```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  Dialogs,
ExtCtrls, StdCtrls;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    Timer1: TTimer;
    Shape1: TShape;
    procedure Timer1Timer(Sender: TObject);
    procedure Shape1ContextPopup(Sender: TObject; MousePos: TPoint;
      var Handled: Boolean);
    procedure Button1Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin

end;

procedure TForm1.Edit1Change(Sender: TObject);
begin

end;

procedure TForm1.Shape1ContextPopup(Sender: TObject; MousePos: TPoint;
  var Handled: Boolean);
begin

end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin

end;

end.
```

5. Поняття форми, елемента управління, події, обробника події. Редагування коду обробника події.

Коли програміст створює в операційному середовищі Windows нову програму, кажуть, що з'являється новий додаток для Windows. Використання слова «додаток» може здатися дещо незвичним, але це буквальний переклад з англійської Windows application.

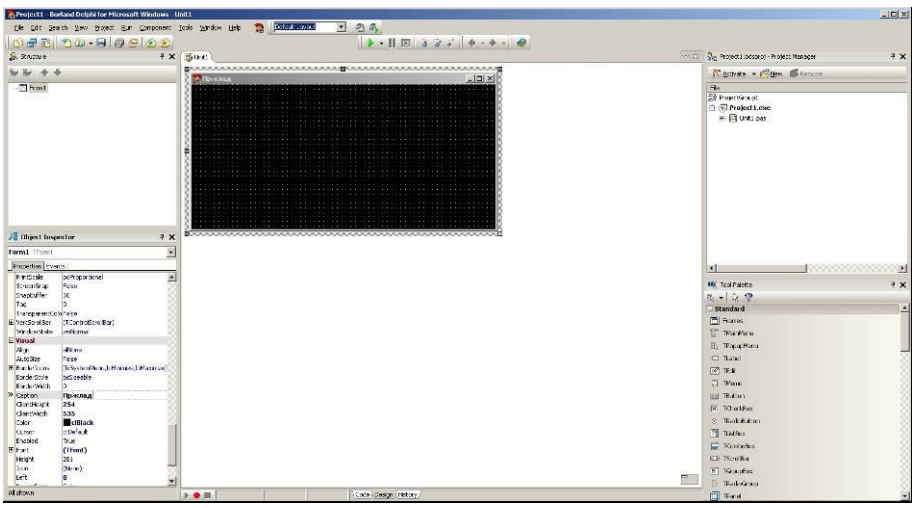
Всі виконувані програми у Windows традиційно поділяються на додатки та служби. Термін «додаток» частіше за все відноситься до великої програми, яка має інтерфейс користувача та широкий вибір функцій. На відміну від додатку, служба – це програма, яка має дуже вузький спектр функціональних можливостей, до яких, як правило, не належать додаткові функції, які мають самостійне значення. Служби зазвичай виконуються у фоновому режимі та мають невеликий інтерфейс користувача або ж не мають його взагалі.



Інтерфейс програми на Delphi розпочинається із форми. Безумовно, існують консольні додатки, які не мають форм, а виконуються в окремому вікні без графіки із введенням / виведенням білими символами на чорному фоні, проте така форма програмування не має нічого спільного із візуальним програмуванням задач. При створенні в Delphi нового додатку (New – Application) перше, що Ви побачите на екрані, - це велика заготовка форми із написом Form1 в заголовку.

Форма являє собою реалізацію вікна операційної системи Windows на об'єктно орієнтованій мові програмування. Властивості та події форми, як й інших компонентів, можна задавати та змінювати за допомогою інспектора об'єктів. Графічний додаток може складатись із декількох форм, але тільки одна із них вважається головною (головна форма відображається на екрані при запуску додатку; закриття головної форми призводить до завершення роботи додатку). Якщо при розміщенні відеокомпонентів на формі постійно перемикати вікна, слідкуючи за змінами програмного коду, то можна побачити, як до тексту програми автоматично додаються нові рядки. Розглянемо, як користуватися вікном інспектора об'єктів на наступному прикладі.

Завдання. При створенні нового додатку на екрані з'являється заготовка форми. Потрібно її зафарбувати в чорний колір та вивести в заголовку форми слово «Приклад».



Розв'язок. У вікні інспектора об'єктів вибираємо властивість Color, поряд із яким

розміщене поле введення для редагування значення параметра (для цього параметру передбачається вибір із меню).

Таким чином, властивість (property) – це елемент компоненту або класу, який відповідає за деяку характеристику елементу чи класу (колір, напис, розмір шрифту тощо). Програмний запис або зчитування інформації про властивість компоненту приводять до виклику пов'язаних із ним процедур та функцій (методів), які в явному виді ми не викликали. Наприклад, при зміні властивості Width (ширина) для деякого візуального компоненту автоматично перемальовується цей компонент (якщо цей компонент – форма, то перемальовується вся форма). Багато властивостей компонентів відображаються в інспекторі об'єктів та можуть бути зміненими у процесі проектування. Доступ до деяких властивостей можливий лише під час виконання програми. Серед них можуть бути властивості тільки для зчитування, безпосереднє змінювання яких у програмі заборонене.

Виберемо в меню поля введення потрібне значення (за умовою задачі – clBlack) і форма автоматично стане чорного кольору. Аналогічним чином виберемо властивість Caption та введемо в рядок значення слово «Приклад» замість слова Form1. В результаті ми отримаємо форму, яка нам потрібна була за умовою задачі.

Змінити значення більшості властивостей можна не лише в процесі проектування, але й під час виконання програми. Це здійснюється за допомогою оператора присвоювання, проте новому додатку необхідно надати сигнал, коли потрібно розпочинати виконання конкретних дій. Такий сигнал називається подією.

В Delphi реалізовано управління програмою як наслідок події. Відразу після запуску програма зупиняється в очікуванні деякої керуючої дії, яка може примусити виконати її ті чи інші команди. Такою подією може бути натискування клавіші на клавіатурі, клацання мишкою, завершення роботи таймера тощо. Кожен компонент на екрані (форма, кнопка, малюнок тощо) налаштований на реакцію тільки на чітко визначені події. Наприклад, малюнок не передбачено для введення тексту, тому він ніяк не реагує на натискування клавіш на клавіатурі.

Для кожної передбаченої події у відеокомпоненту існує метод – процедура, яка описує, що потрібно зробити при настанні події.

Таким чином, подія (event) – це зовнішня керована дія на компонент, яка дозволяє зв'язати його із обробником події – процедурою, яка передбачена для кожної стандартної ситуації (наприклад, при клацанні мишкою на об'єкті чи при отриманні компонентом фокусу). Імена подій розпочинаються із префіксу «On». Події компоненту та пов'язані із ними обробники вказуються на вкладці Events інспектора об'єктів.

Метод (method) – це процедура або функція, включена в опис компоненту (класу). Сукупність методів визначає «дії», які можуть виконувати об'єкти даного класу.

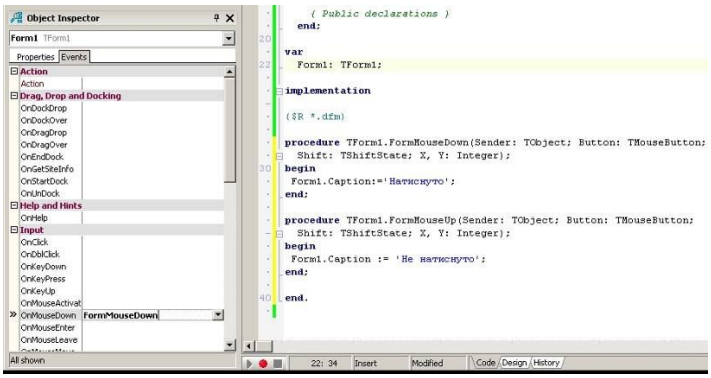
Компонент (component) – це елемент управління відеододатком. Класи компонентів є похідними базового компоненту-предку – класу TComponent. Майже всі компоненти, за винятком тих, які влаштовані в інші компоненти (наприклад, ToolButton або TabSheet), розміщуються у палітрі компонентів і можуть бути перенесені на форму при проектуванні програми.

У назві компоненту як правило нехтують префікс «T», який входить в ім'я відповідного класу: так, компонент типу TEdit називається просто Edit. Серед компонентів виділяють візуальні компоненти (потомки класу TControl), що відображаються на формі під час виконання програми, а серед візуальних – віконні компоненти (потомки класу TWinControl), які можуть отримувати фокус введення та опрацьовувати події, що виникли (наприклад, при натискуванні клавіші та клацанні мишкою).

Розглянемо найпростіший приклад програмування методів для обробки подій.

Завдання. Є порожня форма чорного кольору з написом «Приклад» в заголовку. Потрібно змінити програму таким чином, щоб при натискуванні кнопки мишки (в цей момент мишка повинна знаходитись над формою) в заголовку з'являвся напис «Натиснуто», а при відпусканні кнопки – «Не натиснуто».

Розв'язок. Для програмування цієї задачі потрібно перейти на другу закладку інспектора об'єктів Events та знайти рядок OnMouseDown – це метод, який виконується при натискуванні кнопки мишки.



За допомогою подвійного клацання мишкою на правому полі в рядку OnMouseDown відкриваємо вікно редагування програмового коду. При цьому рядки програми з назвою процедури та її операторними дужками begin … end; вписуються автоматично. Залишається лише дописати рядок програми, за допомогою якого, відповідно до умови задачі, присвоюється нове значення властивості Caption (заголовок) компоненту Form1. Вписуємо цей рядок:

```
procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
 Form1.Caption:='Натиснуто';
end;
```

Аналогічним чином програмуємо метод OnMouseUp:

```
procedure TForm1.FormMouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
 Form1.Caption := 'Не натиснуто';
end;
```

Вітаємо! Ви створили перший додаток для Windows! Запустити програму на виконання можна, натиснувши кнопку F9 (Run) або іконку Виконати 
.

З метою закріплення матеріалу ускладнимо завдання. Допишемо метод, який би виводив в заголовку форми напис «Мишка рухається» при переміщенні мишки над формою. Цей метод – OnMouseMove:

```
procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
```

```
  Form1.Caption:='Мишка рухається';
end;
```

Аналізуючи результат, звернемо увагу на два цікавих моменти:

- Якщо при запуску програми курсор знаходився не над формою, то ми можемо як завгодно рухати мишку, проте напис «Мишка рухається» не з'явиться (це легко пояснити: написаний нами метод відноситься тільки до форми, тому подія OnMouseMove генерується лише тоді, коли курсор заходить на форму);
- Напис «Не натиснута» тепер у заголовку не з'являється (формально працюють обидва методи, але оскільки відпускання кнопки мишки також вважається її рухом, напис «Мишка рухається» встигає з'явитись замість напису «Не натиснута» раніше, ніж ми встигнемо сформувати зримий образ).

6. Теоретичні питання для самоконтролю.

1. Назвіть основні файли, які входять до складу проекту.
2. Що значить значок плюс (хрестик), який розміщується зліва від назви властивостей об'єкту в інспекторі об'єктів?
3. Які файли (з якими розширеннями) з'являються у робочій папці після компіляції проекту?
4. Перерахуйте основні розділи програмного коду модуля.
5. Яка інформація відображається у вікні менеджера проекту (Project Manager)?
6. У якому вікні можна змінити картинку (іконку), яка з'являється у лівому верхньому кутку проекту, що виконується?
7. Які існують обмеження на ім'я модуля, записаного після ключового слова unit?
8. Що відбудеться, якщо при перенесенні проекту із однієї машини на іншу втратиться файл із розширенням dcu?
9. Як виділити на формі скритий або візуально важко відшукуваний відеокомпонент за допомогою вікна Tree View?
10. Як за допомогою вікна Tree View визначити, чи належить дана кнопка контейнеру-панелі?
11. Для який потреб служить панель інструментів Align?
12. Що таке обробник події?
13. Перерахуйте назви груп обробників події за замовчуванням для компонентів Form, Edit, Button, Shape, Timer.
14. Що являє собою додаток? Як створити новий додаток в Delphi?
15. Що таке форма? Назвіть її основні характеристики.
16. Дайте визначення термінам «властивість», «подія», «метод».
17. Що таке компонент?