МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Український державний університет науки і технологій

Кафедра «Технологія машинобудування»

В авторській редакції

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Навчально-методичні рекомендації
до виконання лабораторного практикуму

Частина ІІ

Електронне видання

ДНІПРО
2024

УДК 004.652.5:531/534(076.5)
О 29

Упорядники:
В. С. Гришин, В. М. Карабут

Електронне видання

Схвалено Групою забезпечення якості освітньої програми
«Технологія машинобудування» спеціальності 131 – Прикладна механіка (бакалаврський рівень)
Протокол №5 від 01.03.2024 р.

О 29 Об'єктно-орієнтоване програмування : навчально-методичні рекомендації до виконання лабораторного практикуму / упоряд. В. С. Гришин, В. М. Карабут ; Укр. держ. ун-т науки і технологій. – Електрон. вид. – Дніпро : УДУНТ, 2024. – Ч. ІІ.– 130 с.

Наведені методика і послідовність проведення лабораторних робіт, вимоги до оформлення звіту з лабораторних робіт, запитання з самоконтролю до розділів, рекомендована література.

Призначені для студентів спеціальності 131 – Прикладна механіка (бакалаврський рівень).

ЗМІСТ

РОЗДІЛ 4. МЕТОДИ ОБРОБКИ ДАНИХ

Лабораторна робота №9

Тема: Циклічні структури. Оператор for.

Мета роботи: Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп`ютерного забезпечення в технологічних процесах обробки металів різанням.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №9.
3. Скласти звіт з лабораторної роботи №9 за наведеною формою (див. Додаток І).

Хід виконання роботи

1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

2. Алгоритм виконання лабораторної роботи №9

2.1. Запуск програми Delphi

У системному меню «***Пуск***», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7**.

2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту виберіть в «***Головному***» меню **File** (Файл) **New** (Нова) > **Application** (Програма) (рис. 2.1).

2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 9 – Lab_9).

Рекомендується імені проєкту надавати ім’я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 9 – Project_9).

Файл проєкту Project _9 потрібно зберегти в папці Lab_9.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім’я, як у проєкті (наприклад, Unit _9). Його також потрібно зберегти у папці Lab_9.

На панелі інструментів «***Standard***» (Стандартна) натиснути кнопку  **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

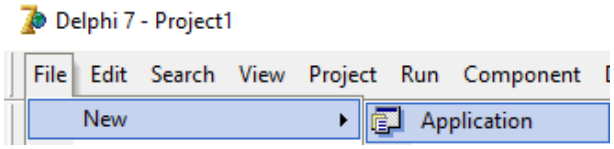


Рис. 2.1. Виберіть **New** > **Application** (Нова > Програма) в «***Головному***» меню **File** (Файл)

Рис. 2.2. Натиснути кнопку  **Save all** (Зберегти все) на панелі інструментів «***Standard***» (Стандартна)

2.4. Розробка програми

Розробити програму для обчислення сили струму в електричному ланцюгу верстата, якщо напруга U змінюється від 220 до 225 В з кроком 1 В, а R = 800 Ом. Відповідно до закону Ома, використовуємо формулу

$$I = U/R$$

2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рис. 2.3. Вивід повідомлень програми представлено на рис. 2.4 (а, б).

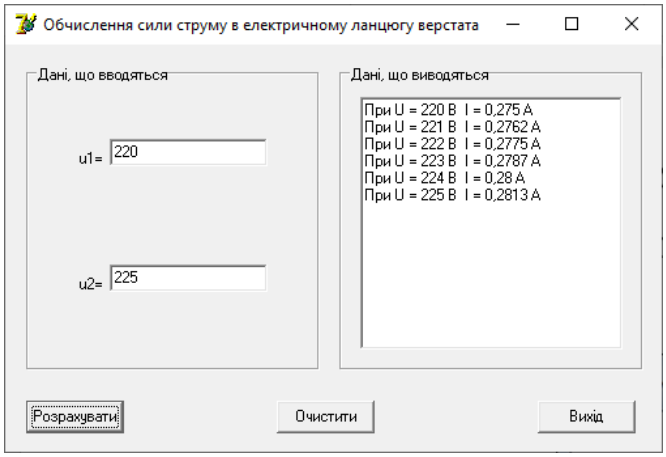
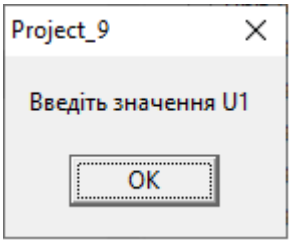
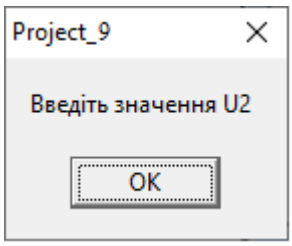


Рис. 2.3. Інтерфейс програми



а



б

Рис. 2.4. Вивід повідомлень програми: а, б

2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «***Form1***» (Форма1) (рис. 2.5). Вікно форми «***Form1***» (Форма1) являє собою шаблон вікна розробляємої програми.

2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «***Form1***» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) на «***Панелі компонентів***» на вкладці «***Standard***» (Стандартна) (рис. 2.7) вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

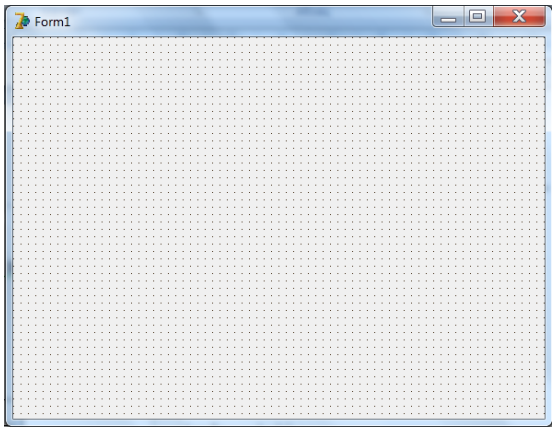


Рис. 2.5. Вікно форми «***Form1***» (Форма1)

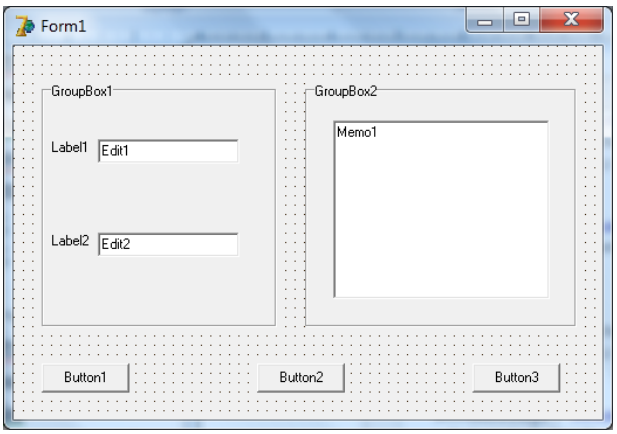


Рис. 2.6. Компоненти розміщені на вікні форми «***Form1***» (Форма1)

б) потім вказати курсором миші у довільному місці на формі «***Form1***» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «***Form1***» (Форма1) (рис. 2.6).

Рис. 2.7. «***Панель компонентів***» на вкладці «***Standard***» (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	**Піктограма**	**Властивість**	**Значення**
Form1	-	Caption	Обчислення сили струму в електричному ланцюгу верстата
		BorderStyle	bsSingle
Groupbox1		Caption	Дані, що вводяться
Groupbox2		Caption	Дані, що виводяться
Edit1		Name	edtInputu1
		Text	значення не вводимо, а тільки очистити поле
Edit2		Name	edtInputu2
		Text	значення не вводимо, а тільки очистити поле
Label1		Caption	u1=
Label2		Caption	u2=

Продовження таблиці 2.1

Компонент	Піктограма	Властивість	Значення
Memo1		Name	memOutput
		Lines	натиснути кнопку ... У з'явившемуся вікні очистити поле, потім натиснути кнопку **ОК**
		ReadOnly	True
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «***Form1***» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

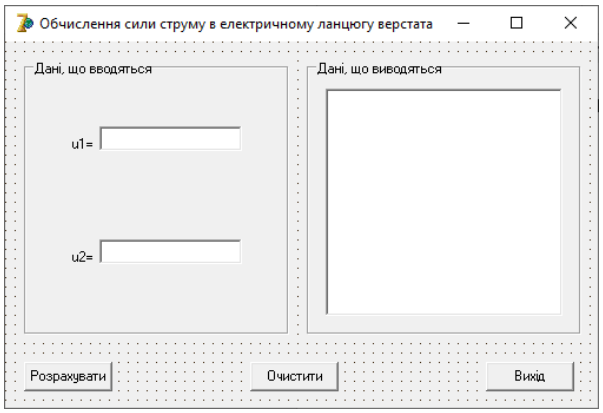


Рис. 2.8. Компоненти та вікно форми «***Form1***» (Форма1) із зміненими властивостями

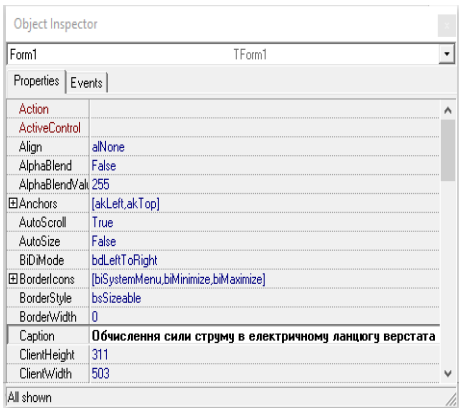


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «***Form1***» (Форма1) у вікні «***Object Inspector***» (Інспектор об'єктів) на **Обчислення сили струму в електричному ланцюгу верстата**

б) у лівій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);

б) у цьому ж вікні потрібно вибрати вкладку «***Events***» (Події) (рис. 2.10);

в) у лівій колонці вкладки «***Events***» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об’єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення обробника події потрібно у правій колонці вкладки «***Events***» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «***Unit_9.pas***» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «***Object Inspector***» (Інспектор об’єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «***Unit_9.pas***» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п 2.4.3, наведені тексти інструкцій обробки події **OnClick** для кнопок: **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit).

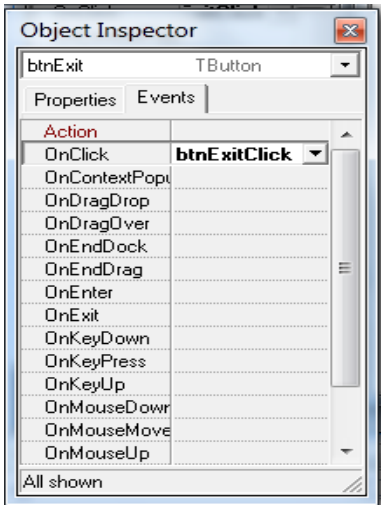


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «***Events***» (Події) у вікні «***Object Inspector***» (Інспектор об'єктів)

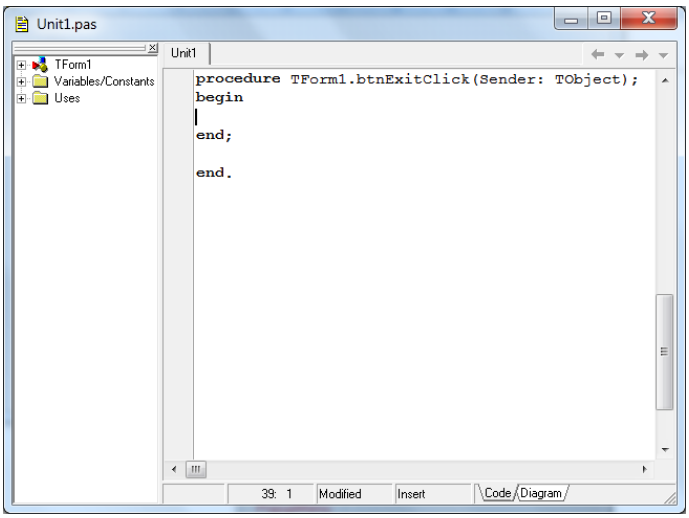


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

Лістинг програми

```
unit Unit_9;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;
type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    edtinputu1: TEdit;
    edtinputu2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
```

```
    memOutput: TMemo;
    btnCalculate: TButton;
    btnClear: TButton;
    btnExit: TButton;
    procedure btnCalculateClick(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1. btnCalculateClick(Sender: TObject);
const
   R : Integer=800; // Оголошення констант
var
   // Оголошення змінних
   U, u1, u2 : integer;
   I : real;
   code : integer;
   tmp : integer;
begin
   // Ввід даних із захистом полів вводу
   // U1
   val(edtInputu1.text,u1,code);
   if code<>0 then
   begin
   showmessage('Введіть значення U1');
   edtInputu1.SetFocus;
   exit;
   end;
```

```
    // U2
   val(edtinputu2.text,u2,code);
   if code<>0 then
   begin
   showmessage('Введіть значення U2');
   edtInputu2.SetFocus;
   exit;
   end;
    // Обмін місцями початкового та кінцевого значення при неправильному
введенні даних
    if u1>u2 then
    begin
      tmp:=u1;
      u1:=u2;
      u2:=tmp;
    end;
   // Очищення полів виводу даних
   memOutput.Lines.Clear;
   // Цикл
   for U:=u1 to u2 do
   begin
   // Обробка даних
   I := U/R;
   // Вивід даних із форматуванням
   memOutput.Lines.Add('При U = ' + IntToStr(U) + ' В' +  '  I = ' +
   FloatToStrF(I,ffGeneral,4,2) + ' А');
   end;
end;
procedure TForm1.btnClearClick(Sender: TObject);
begin
  // Очищення полів вводу даних
  edtinputu1.Text:=' ';
  edtinputu2.Clear;
  // Очищення полів виводу даних
  memOutput.lines.Clear;
```

```
end;
procedure TForm1.btnExitClick(Sender: TObject);
begin
  // Закриття програми
  Application.Terminate;
end;
end.
```

2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

Компіляція програми виконується з «***Головного***» меню **Project** (Проєкт) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

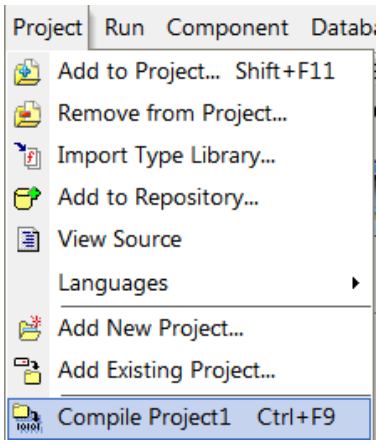


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «***Головного***» меню **Project** (Проєкт)

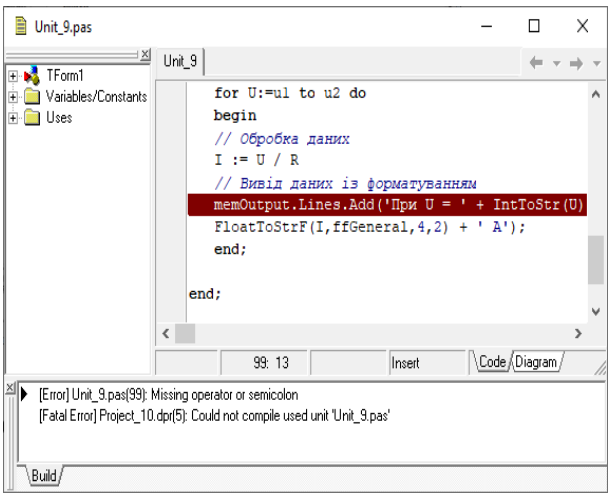


Рис. 2.13. Повідомлення компілятора про виявлені помилки

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно

локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

2.4.5. Запуск програми

Запуск програми може виконуватися з «***Головного***» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «***Debug***» (Налагодження) (рис. 2.14) або натиснути клавішу **F9.**

Рис. 2.14. Панель інструментів «***Debug***» (Налагодження)

2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами винятків є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «***Головного***» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

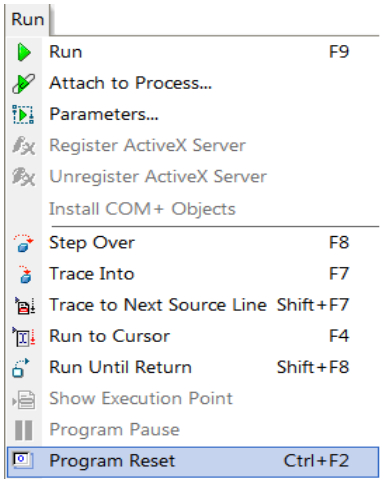


Рис.2.15. Вибір команди **Program Reset** (Скидання програми) із «***Головного***» меню **Run** (Запуск)

2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №9 використовуємо рядкову функцію Val, яка дозволяє використовувати такі види захисту поля вводу (див. п. п. 2.4.3):

- захист поля вводу від вводу порожнього рядка;
- захист поля вводу від вводу цифри «0» нуль;
- захист поля вводу від вводу літер та деяких символів.

3. Зміст звіту з лабораторної роботи №9

Звіт з лабораторної роботи №9 складається з:

- титульного аркуша (див. Додаток І);
- аркуша (див. Додаток І) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть **умову** та **формулу**, які розглядаються в лабораторній роботі №9, див. п. п. 2.4);
- аркуша (див. Додаток І) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №9, див. п. п. 2.4.1);
- аркушів (див. Додаток І) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №9, див. п. п. 2.4.3) та **висновки**.

Лабораторна робота №10

Тема: Циклічні структури. Оператор while.

Мета роботи: Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп`ютерного забезпечення в технологічних процесах обробки металів різанням.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №10.
3. Скласти звіт з лабораторної роботи №10 за наведеною формою (див. Додаток К).

Хід виконання роботи

1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

2. Алгоритм виконання лабораторної роботи №10

2.1. Запуск програми Delphi

У системному меню «***Пуск***», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7**.

2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту в «***Головному***» меню **File** (Файл) вибрати: **New** (Нова) > **Application** (Програма) (рис. 2.1).

2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 10 – Lab_10).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 10 – Project_10).

Файл проєкту Project _10 потрібно зберегти в папці Lab_10.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit _10). Його також потрібно зберегти у папці Lab_10.

На панелі інструментів «***Standard***» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

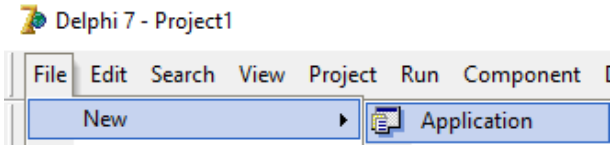


Рис. 2.1. Виберіть **New** > **Application** (Нова > Програма) в «***Головному***» меню **File** (Файл)

Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «***Standard***» (Стандартна)

2.4. Розробка програми

Розробити програму для обчислення значення шорсткості Ra поверхні, обробленою шліфуванням, за формулою:

$$Ra = c \cdot t^z$$

При зміні глибини різання t від 0,01 до 0,04 мм з кроком 0,01 мм, де показник ступеня z = 0,4, а значення коефіцієнта c = 2.

2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рис. 2.3. Вивід повідомлень програми представлено на рис. 2.4 (а, б).

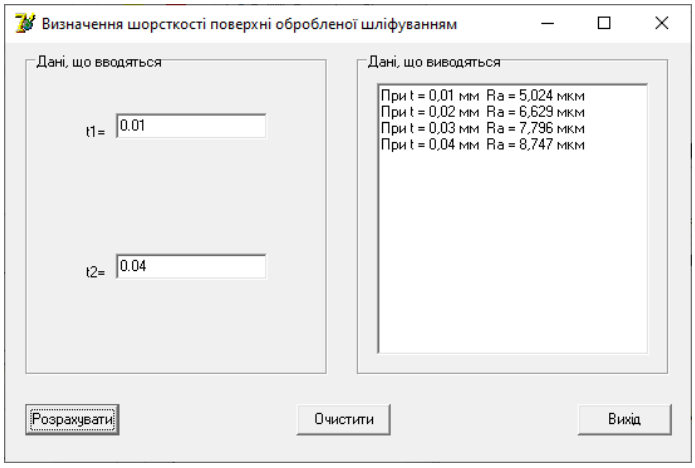
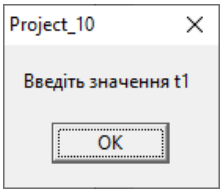
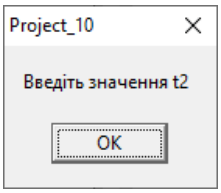


Рис. 2.3. Інтерфейс програми



а



б

Рис. 2.4. Вивід повідомлень програми: а, б

2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «***Form1***» (Форма1) (рис. 2.5). Вікно форми «***Form1***» (Форма1) являє собою шаблон вікна розробляємої програми.

2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «***Form1***» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) на «***Панелі компонентів***» на вкладці «***Standard***» (Стандартна) (рис. 2.7) вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

б) потім вказати курсором миші у довільному місці на формі «***Form1***» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «***Form1***» (Форма1) (рис. 2.6).

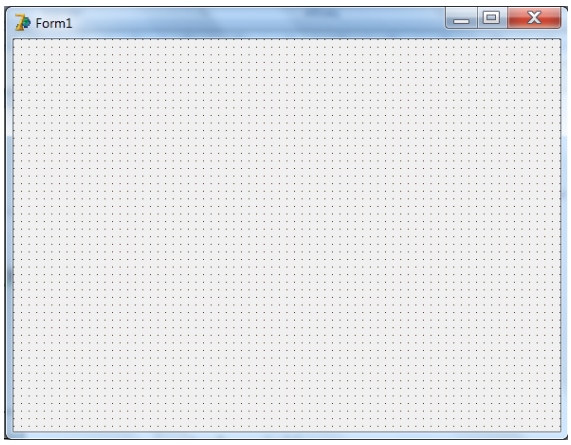


Рис. 2.5. Вікно форми «***Form1***» (Форма1)

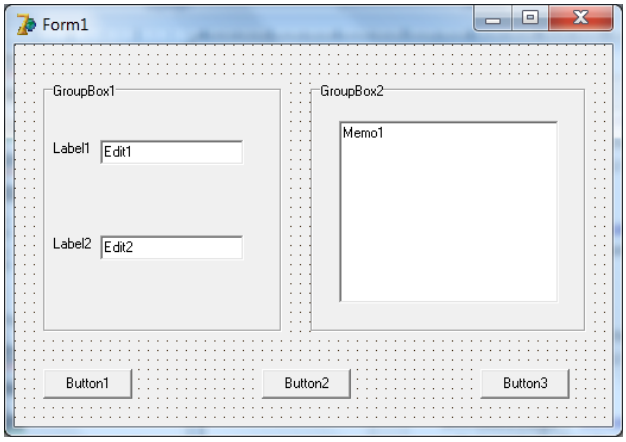


Рис. 2.6. Компоненти розміщені на вікні форми «***Form1***» (Форма1)

Рис. 2.7. «***Панель компонентів***» на вкладці «***Standard***» (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	**Піктограма**	**Властивість**	**Значення**
Form1	-	Caption	Визначення шорсткості поверхні обробленої шліфуванням
		BorderStyle	bsSingle
Groupbox1		Caption	Вводимі дані
Groupbox2		Caption	Виводимі дані
Edit1		Name	edtInputt1
		Text	значення не вводимо, а тільки очистити поле
Edit2		Name	edtInputt2
		Text	значення не вводимо, а тільки очистити поле
Label1		Caption	t1=
Label2		Caption	t2=
Memo1		Name	memOutput
		Lines	натиснути кнопку . У з'явившемуся вікні очистити поле, потім натиснути кнопку **ОК**
		ReadOnly	True
Button1		Name	btnCalculate
		Caption	Розрахувати

Продовження таблиці 2.1

Компонент	Піктограма	Властивість	Значення
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «***Form1***» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

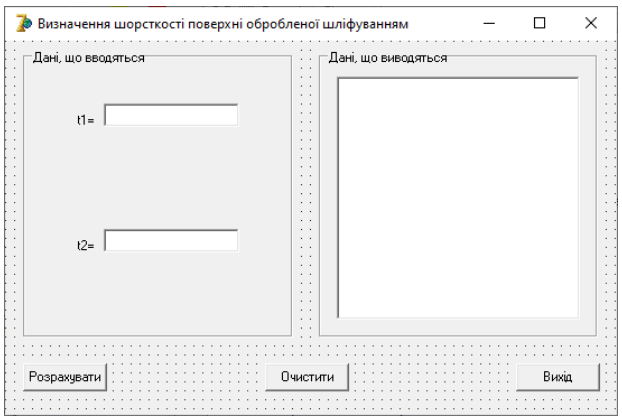


Рис. 2.8. Компоненти та вікно форми «***Form1***» (Форма1) із зміненими властивостями

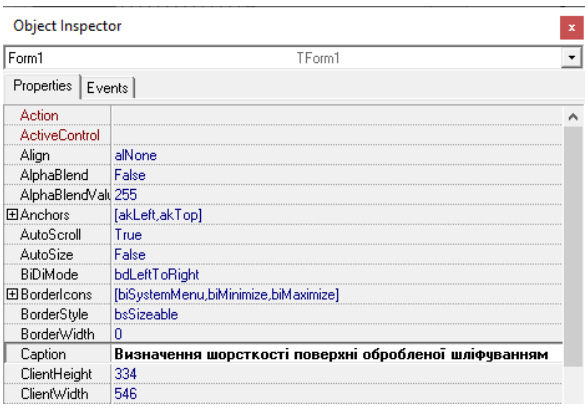


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «***Form1***» (Форма1) у вікні «***Object Inspector***» (Інспектор об'єктів) на **Визначення шорсткості поверхні обробленої шліфуванням**

2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);

б) у цьому ж вікні потрібно вибрати вкладку «***Events***» (Події) (рис. 2.10);

в) у лівій колонці вкладки «***Events***» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об’єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення обробника події потрібно у правій колонці вкладки «***Events***» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

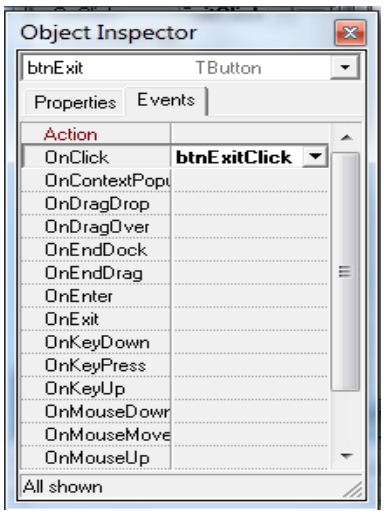


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «***Events***» (Події) у вікні «***Object Inspector***» (Інспектор об'єктів)

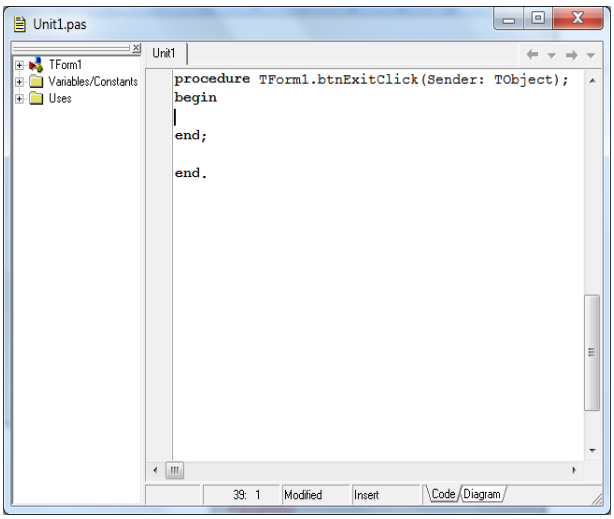


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

д) відкриється вікно «***Unit_10.pas***» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «***Object Inspector***» (Інспектор об’єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «***Unit_10.pas***» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій обробки події **OnClick** для кнопок: **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit).

2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми. Так як у вихідному коді використовуються математичні функції, то необхідно в розділі **uses** додати модуль **Math**.

Лістинг програми

```
unit Unit_10;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Math;
type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    edtinputt1: TEdit;
    edtinputt2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    memOutput: TMemo;
    btnCalculate: TButton;
    btnClear: TButton;
    btnExit: TButton;
```

```
    procedure btnCalculateClick(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1. btnCalculateClick(Sender: TObject);
const
   // Оголошення констант
   c : Integer=2;
   z : real = 0.4;
var
   // Оголошення змінних
   t, t1, t2 : real;
   Ra, tmp : real;
   code : integer;
begin
   // Ввід даних із захистом полів вводу
   // t1
   val(edtInputt1.text,t1,code);
   if code<>0 then
   begin
   showmessage('Введіть значення t1');
   edtInputt1.SetFocus;
   exit;
   end;
   // t2
   val(edtinputt2.text,t2,code);
   if code<>0 then
```

```
  begin
  showmessage('Введіть значення t2');
  edtInputt2.SetFocus;
  exit;
  end;
  // Очищення поля виводу
  memOutput.Lines.Clear;
   // Обмін місцями початкового та кінцевого значення при неправильному
введенні даних
   if t1>t2 then
   begin
     tmp:=t1;
     t1:=t2;
     t2:=tmp;
   end;
  // Цикл
  t:=t1;
  while t<=t2 do
  begin
  // Обробка даних
  Ra := c*power(t*1000,z);
  // Вивід даних із форматуванням
  memOutput.Lines.Add('При t = ' + FloatToStr(t) + ' мм' +  '  Ra = ' +
  FloatToStrF(Ra,ffGeneral,4,2) + ' мкм');
  // Крок циклу
  t:=t+0.01;
  end;
end;
procedure TForm1.btnClearClick(Sender: TObject);
begin
 // Очищення полів вводу даних
 edtinputt1.Text:=' ';
 edtinputt2.Clear;
 // Очищення полів виводу даних
 memOutput.lines.Clear;
```

```
end;
procedure TForm1.btnExitClick(Sender: TObject);
begin
  // Закриття програми
  Application.Terminate;
end;
end.
```

2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

Компіляція програми виконується з «***Головного***» меню **Project** (Проєкт) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

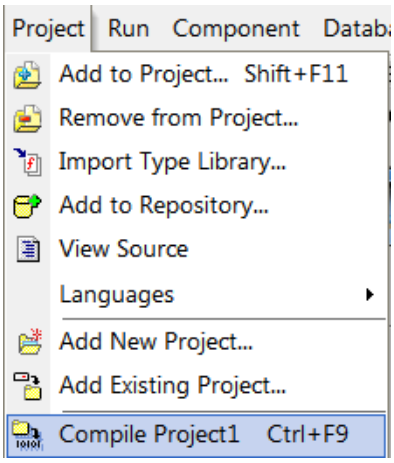


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «***Головного***» меню **Project** (Проєкт)

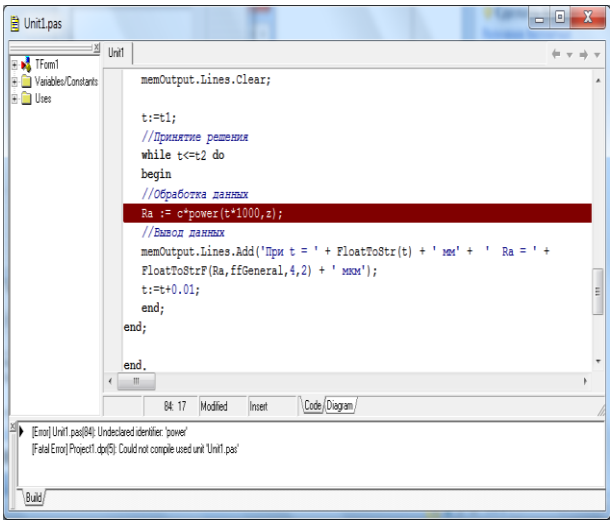


Рис. 2.13. Повідомлення компілятора про виявлені помилки

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно

локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

2.4.5. Запуск програми

Запуск програми може виконуватися з «***Головного***» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) панелі інструментів «***Debug***» (Налагодження) (рис. 2.14) або натиснути клавішу **F9.**

Рис. 2.14. Панель інструментів «***Debug***» (Налагодження)

2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами винятків є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «***Головного***» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

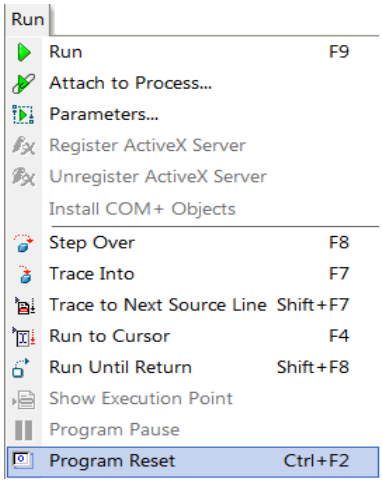


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «***Головного***» меню **Run** (Запуск)

2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №10 використовуємо рядкову функцію Val, яка дозволяє використовувати такі види захисту поля вводу (див. п. п. 2.4.3):

- захист поля вводу від вводу порожнього рядка;
- захист поля вводу від вводу цифри «0» нуль;
- захист поля вводу від вводу літер та деяких символів.

3. Зміст звіту з лабораторної роботи №10

Звіт з лабораторної роботи №10 складається з:

- титульного аркуша (див. Додаток К);
- аркуша (див. Додаток К) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть **умову** та **формулу**, які розглядаються в лабораторній роботі №10, див. п. п. 2.4);
- аркуша (див. Додаток К) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №10, див. п. п. 2.4.1);
- аркушів (див. Додаток К) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №10, див. п. п. 2.4.3), **висновки**.

Лабораторна робота №11

Тема: Циклічні структури. Оператор repeat.

Мета роботи: Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп`ютерного забезпечення в технологічних процесах обробки металів різанням.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №11.
3. Скласти звіт з лабораторної роботи №11 за наведеною формою (див. Додаток Л).

Хід виконання роботи

1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

2. Алгоритм виконання лабораторної роботи №11

2.1. Запуск програми Delphi

У системному меню «***Пуск***», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7**.

2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту в «***Головному***» меню **File** (Файл) вибрати: **New** (Нова) > **Application** (Програма) (рис. 2.1).

2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 11 – Lab_11).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 11 – Project_11).

Файл проєкту Project _11 потрібно зберегти в папці Lab_11.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit _11). Його також потрібно зберегти у папці Lab_11.

На панелі інструментів **Standard** (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

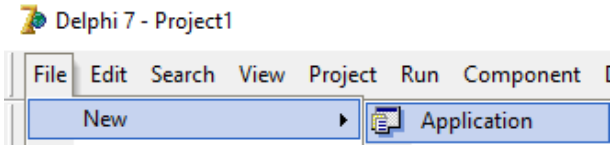


Рис. 2.1. Виберіть **New** > **Application** (Нова > Програма) в «***Головному***» меню **File** (Файл)

Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів **Standard** (Стандартна)

2.4. Розробка програми

Розробити програму для обчислення значення шорсткості Ra поверхні, обробленою шліфуванням, за формулою:

$$Ra = c \cdot t^z$$

При зміні глибини різання t від 0,01 до 0,04 мм з кроком 0,01 мм, де показник ступеня z = 0,4, а значення коефіцієнта c = 2.

2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рис. 2.3. Вивід повідомлень програми представлено на рис. 2.4 (а, б).

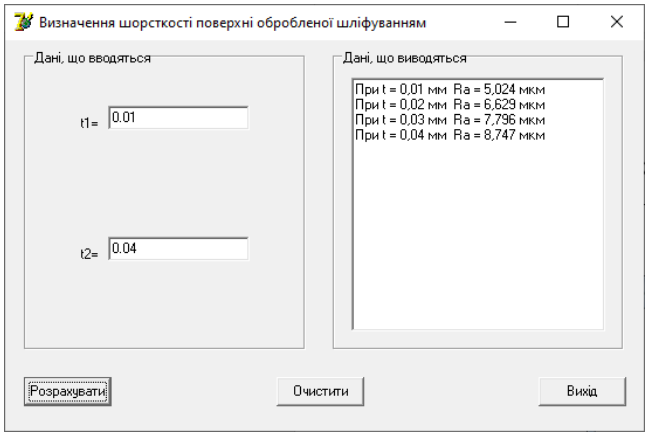
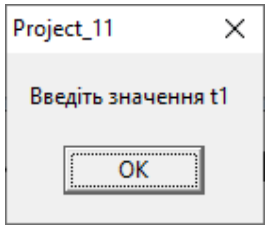
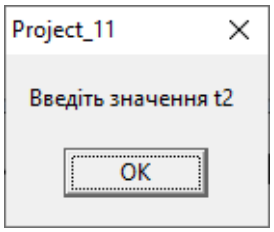


Рис. 2.3. Інтерфейс програми



а



б

Рис. 2.4. Вивід повідомлень програми: а, б

2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «***Form1***» (Форма1) (рис. 2.5). Вікно форми «***Form1***» (Форма1) являє собою шаблон вікна розробляємої програми.

2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «***Form1***» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) на «***Панелі компонентів***» на вкладці «***Standard***» (Стандартна) (рис. 2.7), вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

б) потім вказати курсором миші у довільному місці на формі «***Form1***» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «***Form1***» (Форма1) (рис. 2.6).

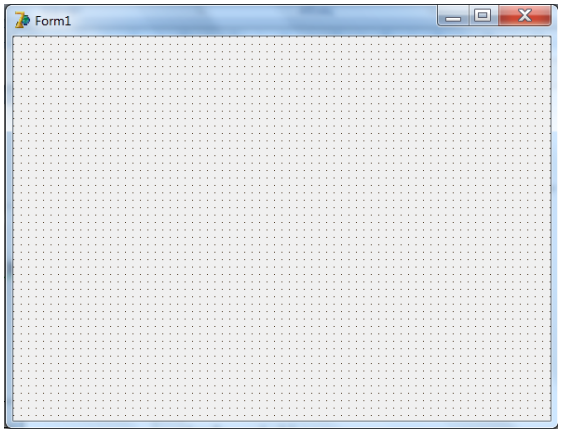


Рис. 2.5. Вікно форми «***Form1***» (Форма1)

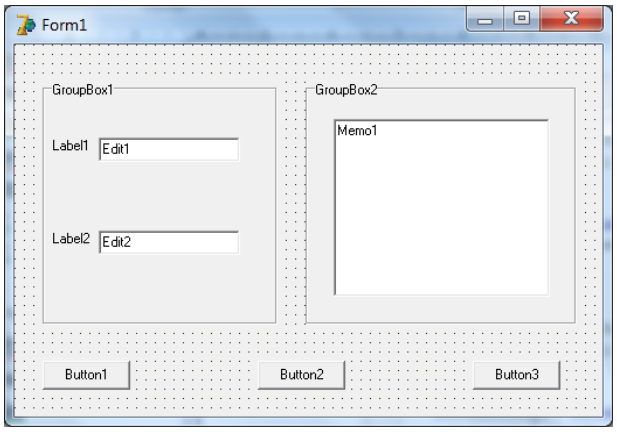


Рис. 2.6. Компоненти розміщені на вікні форми «***Form1***» (Форма1)

Рис. 2.7. «***Панель компонентів***» на вкладці **Standard** (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	**Піктограма**	**Властивість**	**Значення**
Form1	-	Caption	Визначення шорсткості поверхні обробленої шліфуванням
		BorderStyle	bsSingle
Groupbox1		Caption	Дані, що вводяться
Groupbox2		Caption	Дані, що виводяться
Edit1		Name	edtInputt1
		Text	значення не вводимо, а тільки очистити поле
Edit2		Name	edtInputt2
		Text	значення не вводимо, а тільки очистити поле
Label1		Caption	t1=
Label2		Caption	t2=
Memo1		Name	memOutput
		Lines	натиснути кнопку ... У з'явившемуся вікні очистити поле, потім натиснути кнопку **ОК**
		ReadOnly	True

Продовження таблиці 2.1

Компонент	Піктограма	Властивість	Значення
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «***Form1***» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

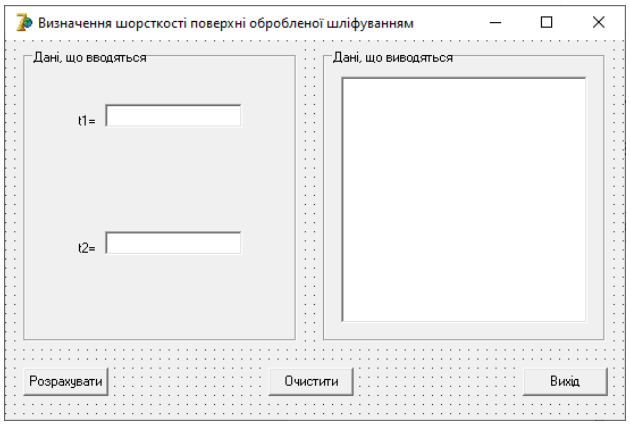


Рис. 2.8. Компоненти та вікно форми «***Form1***» (Форма1) із зміненими властивостями

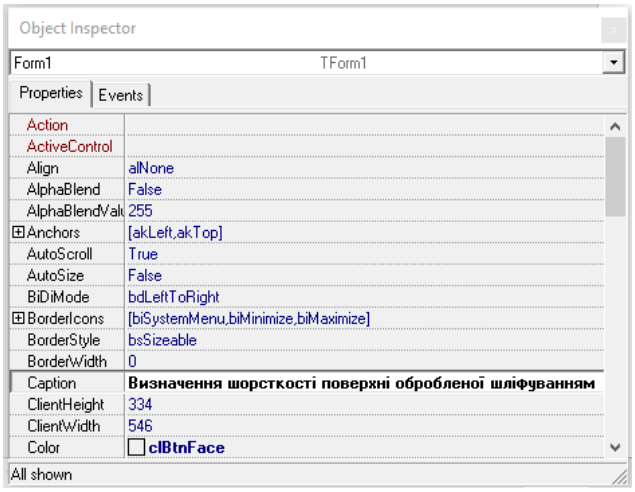


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «***Form1***» (Форма1) у вікні «***Object Inspector***» (Інспектор об'єктів) на **Визначення шорсткості поверхні обробленої шліфуванням**

в) у правій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);

б) у цьому ж вікні потрібно вибрати вкладку «***Events***» (Події) (рис. 2.10);

в) у лівій колонці вкладки «***Events***» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об’єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення обробника події потрібно у правій колонці вкладки «***Events***» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «***Unit_11.pas***» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «***Object Inspector***» (Інспектор об’єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «***Unit_11.pas***» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п 2.4.3, наведені тексти інструкцій обробки події **OnClick** для кнопок: **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit).

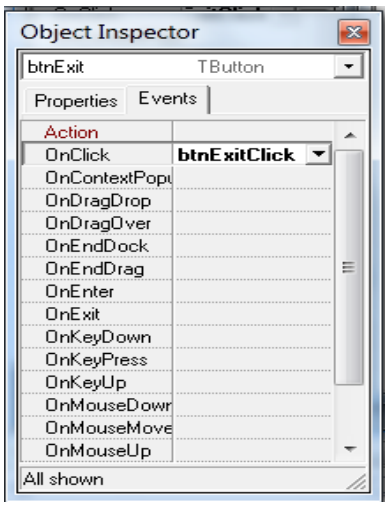


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «***Events***» (Події) у вікні «***Object Inspector***» (Інспектор об'єктів)

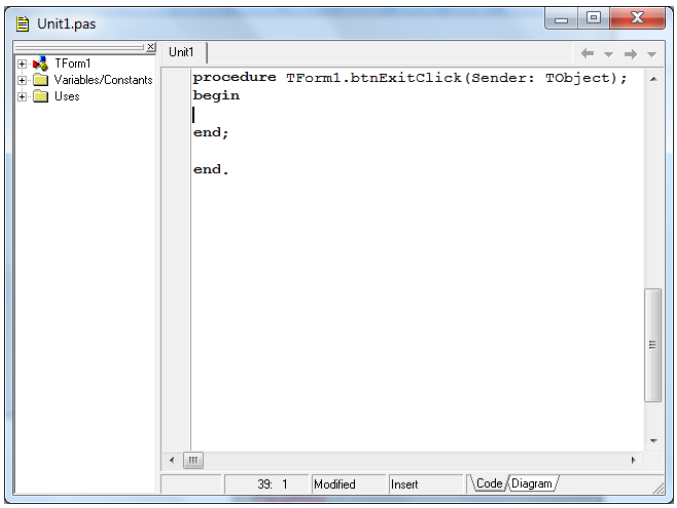


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми. Так як у вихідному коді використовуються математичні функції, то необхідно в розділі **uses** додати модуль **Math**.

Лістинг програми

```
unit Unit _11;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Math;
type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    edtinputt1: TEdit;
    edtinputt2: TEdit;
```

```
    Label1: TLabel;
    Label2: TLabel;
    memOutput: TMemo;
    btnCalculate: TButton;
    btnClear: TButton;
    btnExit: TButton;
    procedure btnCalculateClick(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1. btnCalculateClick(Sender: TObject);
const
 // Оголошення констант
   c : Integer=2;
   z : real = 0.4;
var
   // Оголошення змінних
   t, t1, t2 : real;
   Ra : real;
   code : integer;
   tmp : real;
begin
   // Ввід даних із захистом полів вводу
   // t1
   val(edtInputt1.text,t1,code);
   if code<>0 then
   begin
```

```
  showmessage('Введіть значення t1');
  edtInputt1.SetFocus;
  exit;
  end;
  // t2
  val(edtinputt2.text,t2,code);
  if code<>0 then
  begin
  showmessage('Введіть значення t2');
  edtInputt2.SetFocus;
  exit;
  end;
  // Очищення поля виводу
  memOutput.Lines.Clear;
   // Обмін місцями початкового та кінцевого значення при неправильному
введенні даних
   if t1>t2 then
   begin
     tmp:=t1;
     t1:=t2;
     t2:=tmp;
   end;
   // Цикл
  t:=t1;
  repeat
  // Обробка даних
  Ra := c*power(t*1000,z);
  // Вивід даних із форматуванням
  memOutput.Lines.Add('При t = ' + FloatToStr(t) + ' мм' +  '  Ra = ' +
  FloatToStrF(Ra,ffGeneral,4,2) + ' мкм');
  // Крок циклу
  t:=t+0.01;
  until t>t2;
end;
procedure TForm1.btnClearClick(Sender: TObject);
```

```
begin
  // Очищення полів вводу
  edtinputt1.Text:='';
  edtinputt2.Clear;
  // Очищення полів виводу
  memOutput.lines.Clear;
end;
procedure TForm1.btnExitClick(Sender: TObject);
begin
   // Закриття програми
   Application.Terminate;
end;
end.
```

2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

Компіляція програми виконується з «***Головного***» меню **Project** (Проєкт) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

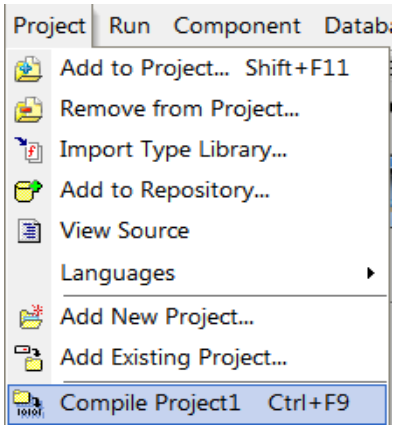


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «***Головного***» меню **Project** (Проєкт)

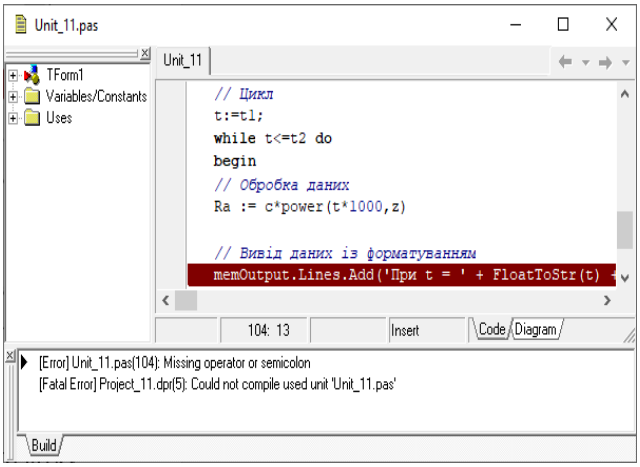


Рис. 2.13. Повідомлення компілятора про виявлені помилки

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

2.4.5. Запуск програми

Запуск програми може виконуватися з «***Головного***» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «***Debug***» (Налагодження) (рис. 2.14) або натиснути клавішу **F9.**

Рис. 2.14. Панель інструментів «***Debug***» (Налагодження)

2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами винятків є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «***Головного***» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

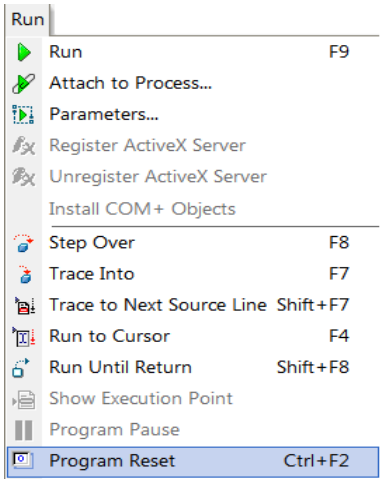


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «***Головного***» меню **Run** (Запуск)

2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №11 використовуємо рядкову функцію Val, яка дозволяє використовувати такі види захисту поля вводу (див. п. п. 2.4.3):

- захист поля вводу від вводу порожнього рядка;
- захист поля вводу від вводу цифри «0» нуль;
- захист поля вводу від вводу літер та деяких символів.

3. Зміст звіту з лабораторної роботи №11

Звіт з лабораторної роботи №11 складається з:

- титульного аркуша (див. Додаток Л);
- аркуша (див. Додаток Л) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть **умову** та **формулу**, які розглядаються в лабораторній роботі №11, див. п. п. 2.4);
- аркуша (див. Додаток Л) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №11, див. п. п. 2.4.1);
- аркушів (див. Додаток Л) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №11, див. п. п. 2.4.3) та **висновки**.

Лабораторна робота №12

Тема: Циклічні структури. Вкладені оператори.

Мета роботи: Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп`ютерного забезпечення в технологічних процесах обробки металів різанням.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №12.
3. Скласти звіт з лабораторної роботи №12 за наведеною формою (див. Додаток М).

Хід виконання роботи:

1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

2. Алгоритм виконання лабораторної роботи №12

2.1. Запуск програми Delphi

У системному меню «***Пуск***», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7**.

2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту, виберіть в «***Головному»*** меню **File** (Файл) команду **New** > **Application** (Нова > Програма) (рис. 2.1).

2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 12 – Lab_12).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 12 – Project_12).

Файл проєкту Project _12 потрібно зберегти в папці Lab_12.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit _12). Його також потрібно зберегти у папці Lab_12.

На панелі інструментів «***Standard***» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

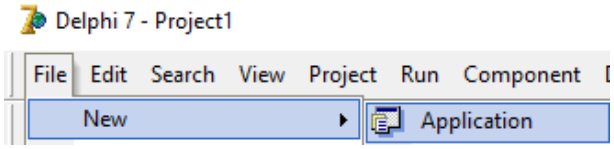


Рис. 2.1. Виберіть **New** > **Application** (Нова > Програма) в «***Головному***» меню **File** (Файл)

Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «***Standard***» (Стандартна)

2.4. Розробка програми

Розробити програму для обчислення значення шорсткості Ra поверхні, обробленою шліфуванням, за формулою

$$Ra = c \cdot t^z$$

При зміні глибини різання t від 0,01 до 0,04 мм з кроком 0,01 мм, відповідно змінюється показник ступеня z від 0,4 до 0,6 з кроком 0,05. Значення коефіцієнта c = 2.

2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рис. 2.3. Вивід повідомлень програми представлено на рис. 2.4 (а, б).

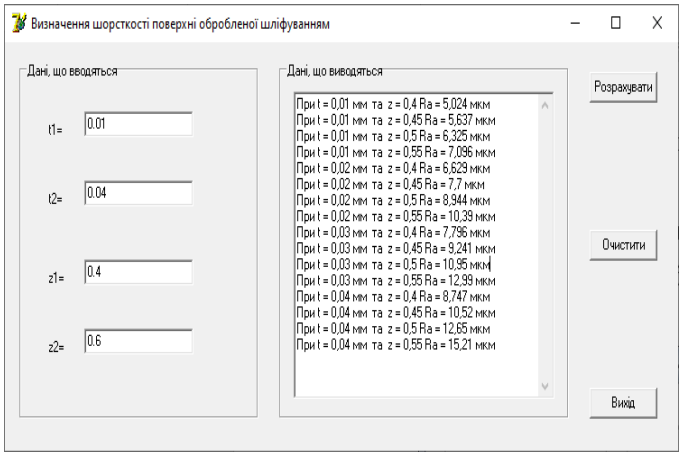


Рис. 2.3. Інтерфейс програми

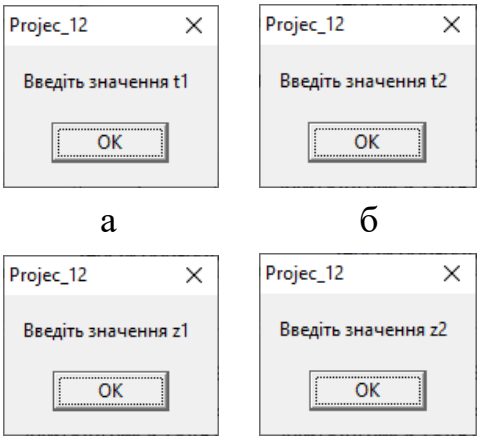


Рис. 2.4. Вивід повідомлень програми: а, б, в, г

2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «***Form1***» (Форма1) (рис. 2.5). Вікно форми «***Form1***» (Форма1) являє собою шаблон вікна розробляємої програми.

2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «***Form1***» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) на «***Панелі компонентів***» на вкладці «***Standard***» (Стандартна) (рис. 2.7) вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

б) потім вказати курсором миші у довільному місці на формі «***Form1***» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «***Form1***» (Форма1) (рис. 2.6).

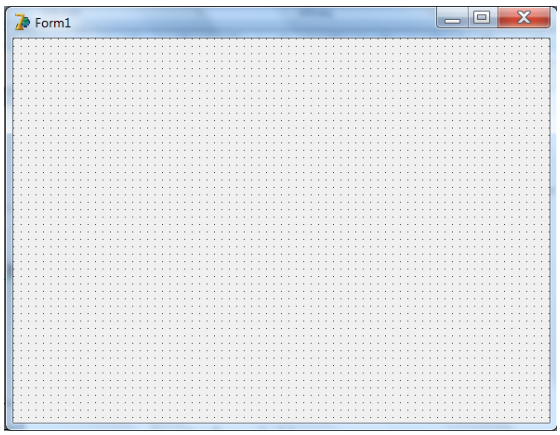


Рис. 2.5. Вікно форми «***Form1***» (Форма1)

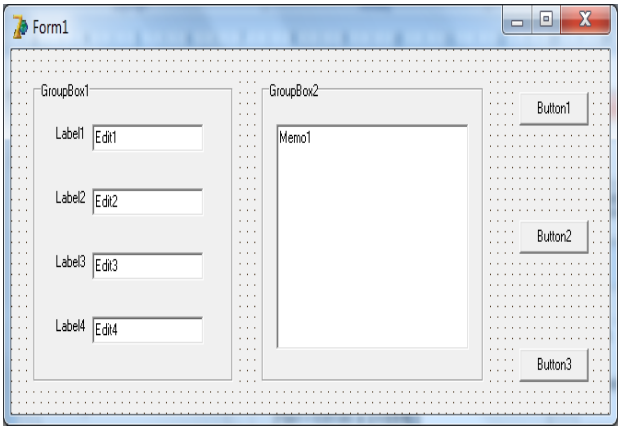


Рис. 2.6. Компоненти розміщені на вікні форми «***Form1***» (Форма1)

Рис. 2.7. «***Панель компонентів***» на вкладці **Standard** (Стандартна)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	**Піктограма**	**Властивість**	**Значення**
Form1	-	Caption	Визначення шорсткості поверхні обробленої шліфуванням
		BorderStyle	bsSingle
Groupbox1		Caption	Дані, що вводяться
Groupbox2		Caption	Дані, що виводяться
Edit1	ab	Name	edtInputt1
		Text	значення не вводимо, а тільки очистити поле
Edit2	ab	Name	edtInputt2
		Text	значення не вводимо, а тільки очистити поле
Edit3	ab	Name	edtInputz1
		Text	значення не вводимо, а тільки очистити поле
Edit4	ab	Name	edtInputz2
		Text	значення не вводимо, а тільки очистити поле
Label1	A	Caption	t1=
Label2	A	Caption	t2=
Label3	A	Caption	z1=
Label4	A	Caption	z2=

Продовження таблиці 2.1

Компонент	Піктограма	Властивість	Значення
Memo1		Name	memOutput
		ScrollBars	ssVertical
		Lines	натиснути кнопку ... У з'явившемуся вікні очистити поле, потім натиснути кнопку кнопку **ОК**
		ReadOnly	True
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «***Form1***» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

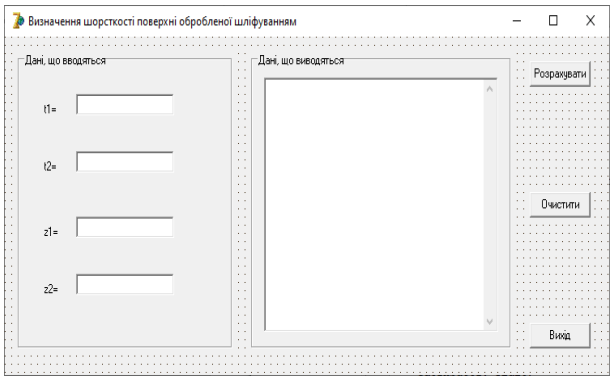


Рис. 2.8. Компоненти та вікно форми «***Form1***» (Форма1) із зміненими властивостями

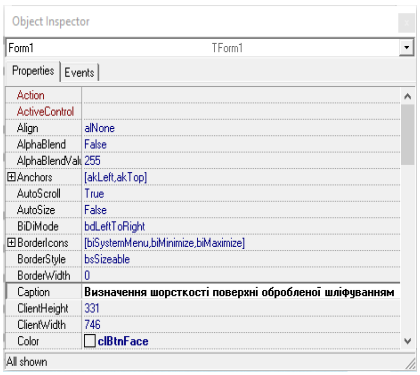


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «***Form1***» (Форма1) у вікні «***Object Inspector***» (Інспектор об'єктів) на **Визначення шорсткості поверхні обробленої шліфуванням**

б) у лівій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);

б) у цьому ж вікні потрібно вибрати вкладку «***Events***» (Події) (рис. 2.10);

в) у лівій колонці вкладки «***Events***» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об’єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення обробника події потрібно у правій колонці вкладки «***Events***» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «***Unit_12.pas***» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «***Object Inspector***» (Інспектор об’єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «***Unit_12.pas***» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п 2.4.3, наведені тексти інструкцій обробки події **OnClick** для кнопок: **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit).

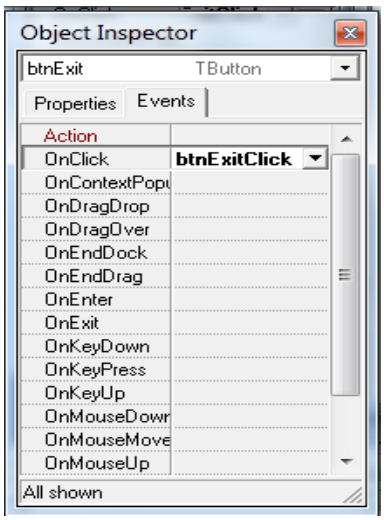


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «***Events***» (Події) у вікні «***Object Inspector***» (Інспектор об'єктів)

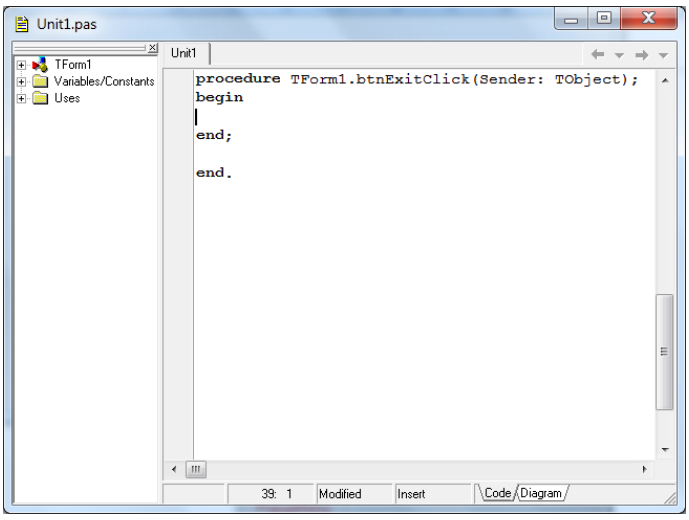


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми. Так як у вихідному коді використовуються математичні функції, то необхідно в розділі **uses** додати модуль **Math**.

Лістинг програми

```
unit Unit_12;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Math;
type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    edtinputt1: TEdit;
    edtinputt2: TEdit;
```

```
    edtinputz1: TEdit;
    edtinputz2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    memOutput: TMemo;
    btnCalculate: TButton;
    btnClear: TButton;
    btnExit: TButton;
    procedure btnCalculateClick(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1. btnCalculateClick(Sender: TObject);
const
  // Оголошення констант
  c : Integer = 2;
var
  // Оголошення змінних
  Ra, t, t1, t2 : Real;
  tmp, z, z1, z2 : Real;
  code : integer;
begin
   // Ввід даних із захистом полів вводу
   // t1
  val(edtinputt1.text,t1,code);
```

```
  if code<>0 then
  begin
  showmessage('Введіть значення t1');
  edtinputt1.SetFocus;
  exit;
  end;
  // t2
  val(edtinputt2.text,t2,code);
  if code<>0 then
  begin
  showmessage('Введіть значення t2');
  edtinputt2.SetFocus;
  exit;
  end;
  // z1
  val(edtinputz1.text,z1,code);
  if code<>0 then
  begin
  showmessage('Введіть значення z1');
  edtinputz1.SetFocus;
  exit;
  end;
  // z2
  val(edtinputz2.text,z2,code);
  if code<>0 then
  begin
  showmessage('Введіть значення z2');
  edtinputz2.SetFocus;
  exit;
  end;
  // Очищення поля виводу
  memOutput.Lines.Clear;
  // Обмін місцями початкового t1 та кінцевого t2 значення при невірному вводі
даних
   if t1>t2 then
```

```
    begin
      tmp:=t1;
      t1:=t2;
      t2:=tmp;
    end;
   // Обмін місцями початкового z1 та кінцевого z2 значення при невірному
вводі даних
    if z1>z2 then
    begin
      tmp:=z1;
      z1:=z2;
      z2:=tmp;
    end;
   // Цикл зовнішній
    t:=t1;
    while t<=t2 do
    begin
          // Цикл внутрішній
          z:=z1;
          while z<=z2 do
          begin
          // Обробка даних
          Ra := c*Power(t*1000,z);
          // Вивід даних з форматуванням
          memOutput.Lines.Add('При t = ' + FloatToStr(t)+ ' мм ' + ' та ' +
          ' z = ' + FloatToStr(z) + ' Ra = ' + FloatToStrF(Ra,ffGeneral,4,2) + ' мкм');
          // Крок внутрішнього циклу
          z:=z+0.05;
          end;
   // Крок зовнішнього циклу
   t:=t+0.01;
   end;
end;
procedure TForm1.btnClearClick(Sender: TObject);
begin
```

```
  // Очищення полів вводу
  edtinputt1.Text:=' ';
  edtinputt2.Clear;
  edtinputz1.Text:=' ';
  edtinputz2.Clear;
  // Очищення полів виводу
  memOutput.lines.Clear;
end;
procedure TForm1.btnExitClick(Sender: TObject);
begin
   // Закриття програми
   Application.Terminate;
end;
end.
```

2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

Компіляція програми виконується з «***Головного***» меню **Project** (Проєкт) обравши команду **Compile Project** (Компіляція проєкту) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу

проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

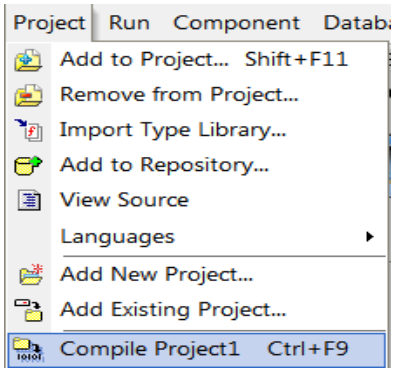


Рис. 2.12. Вибір команди **Compile Project** (Компіляція проєкту) з «***Головного***» меню **Project** (Проєкт)

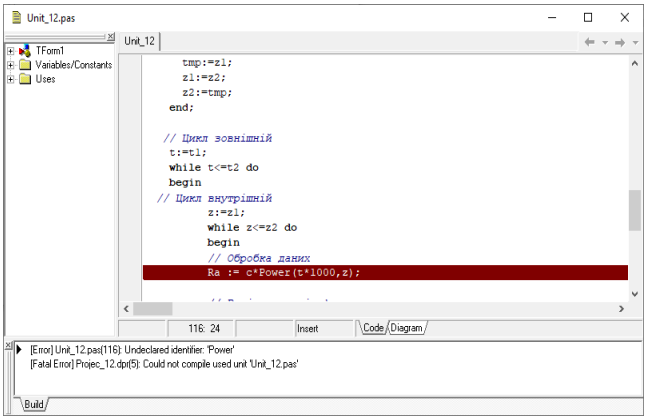


Рис. 2.13. Повідомлення компілятора про виявлені помилки

2.4.5. Запуск програми

Запуск програми може виконуватися з «***Головного***» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «***Debug***» (Налагодження) (рис. 2.14) або натиснути клавішу **F9.**

Рис. 2.14. Панель інструментів «***Debug***» (Налагодження)

2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами винятків є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «***Головного***» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

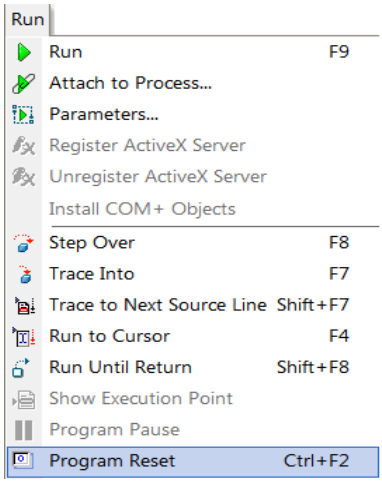


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «***Головного***» меню **Run** (Запуск)

2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №12 використовуємо рядкову функцію **Val**, яка дозволяє використовувати такі види захисту поля вводу (див. п. п. 2.4.3):

- захист поля вводу від вводу порожнього рядка;
- захист поля вводу від вводу цифри «0» нуль;
- захист поля вводу від вводу літер та деяких символів.

3. Зміст звіту з лабораторної роботи №12

Звіт з лабораторної роботи №12 складається з:

- титульного аркуша (див. Додаток М);
- аркуша (див. Додаток М) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть **умову** та **формулу**, які розглядаються в лабораторній роботі №12, див. п. п. 2.4);
- аркуша (див. Додаток М) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №12, див. п. п. 2.4.1);
- аркушів (див. Додаток М) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №12, див. п. п. 2.4.3) та **висновки**.

Запитання з самоконтролю до розділу 4

1. Що таке цикл?
2. Що таке детермінований цикл?
3. Що таке недетермінований цикл?
4. Коли використовується цикл repeat?
5. Коли використовується цикл while?
6. Коли використовується цикл for?
7. У чому суттєва відмінність циклу while від циклу repeat?

РОЗДІЛ 5. СТРУКТУРНЕ ПРОГРАМУВАННЯ

Лабораторна робота №13

Тема: «Структури даних. Одномірний статичний масив»

Мета роботи: Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп`ютерного забезпечення в технологічних процесах обробки металів різанням.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №13.
3. Скласти звіт з лабораторної роботи №13 за наведеною формою (див. Додаток Н).

Хід виконання роботи:

1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

2. Алгоритм виконання лабораторної роботи №13

2.1. Запуск програми Delphi

У системному меню «***Пуск***», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7**.

2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма). Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом. Для створення нового проєкту в «***Головному»*** меню **File** (Файл) вибрати: **New** (Нова) > **Application** (Програма) (рис. 2.1).

2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 13 – Lab_13).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 13 – Project_13).

Файл проєкту Project _13 потрібно зберегти в папці Lab_13.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit _13). Його також потрібно зберегти у папці Lab_13.

На панелі інструментів «***Standard***» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис.2.2).

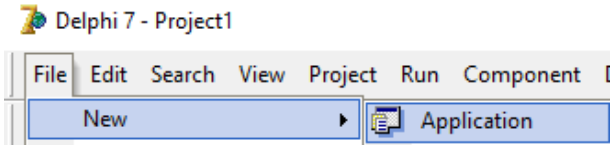


Рис. 2.1. Виберіть **New** > **Application** (Нова > Програма) в «***Головному***» меню **File** (Файл)

Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «***Standard***» (Стандартна)

2.4. Розробка програми

Розробити програму для обчислення кількість проданих за рік автомобілів однієї марки, максимальну та мінімальну кількість проданих автомобілів однієї марки за рік, а також скільки продали автомобілів однієї марки в середньому за рік. Одна марка автомобіля продається поквартально по-різному. Марку автомобіля та кількість проданих автомобілів поквартально прийняти самостійно.

2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рис. 2.3. Вивід повідомлення програми представлено на рис. 2.4.

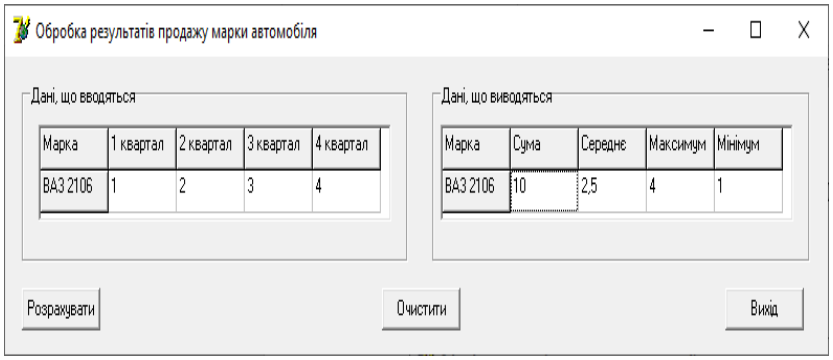


Рис. 2.3. Інтерфейс програми

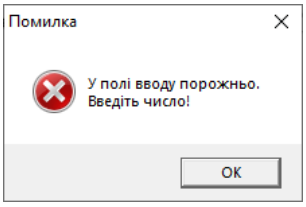


Рис. 2.4. Вивід повідомлення програми

2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «***Form1***» (Форма1) (рис. 2.5). Вікно форми «***Form1***» (Форма1) являє собою шаблон вікна розробляємої програми.

2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «***Form1***» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

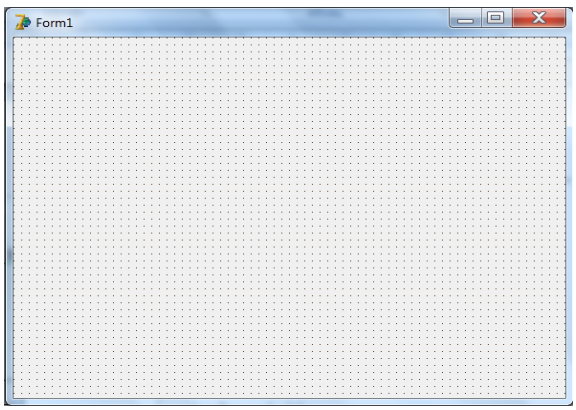


Рис. 2.5. Вікно форми «***Form1***» (Форма1)

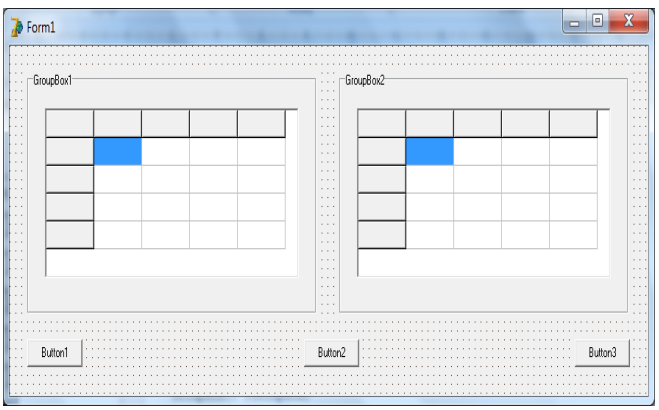


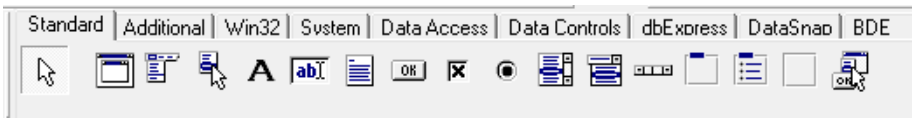
Рис. 2.6. Компоненти розміщені на вікні форми «***Form1***» (Форма1)

Для цього необхідно виконати наступні дії:

а) на «***Панелі компонентів***» на вкладці «***Standard***» (Стандартна) (рис. 2.7, а) та на вкладці «***Additional***» (Додатково) (рис. 2.7, б), вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

б) потім вказати курсором миші у довільному місці на формі «***Form1***» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «***Form1***» (Форма1) (рис. 2.6).



а

б

Рис. 2.7. «***Панель компонентів***»: а) вкладка «***Standard***» (Стандартна), б) вкладка «***Additional***» (Додатково)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	**Піктограма**	**Властивість**	**Значення**
Form1	-	Caption	Обробка результатів продажу марки автомобіля
		BorderStyle	bsSingle
		Name	frmRezult
Groupbox1		Caption	Дані, що вводяться
Groupbox2		Caption	Дані, що виводяться
StringGrid1		Name	GridInput
		ColCount	5
		RowCount	2
		FixedCols	1
		FixedRows	1
		Options:	Розкрити список, натиснувши на знак «+»
		goEditing	True
		goTabs	True
		goAlwaysShowEditing	True

Продовження таблиці 2.1

Компонент	Піктограма	Властивість	Значення
StringGrid2		Name	GridOutput
		ColCount	5
		RowCount	2
		FixedCols	1
		FixedRows	1
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «***Form1***» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1. Увага! Найменування колонок та рядків компонентів StringGrid1 та StringGrid2 можливе лише у вихідному коді.

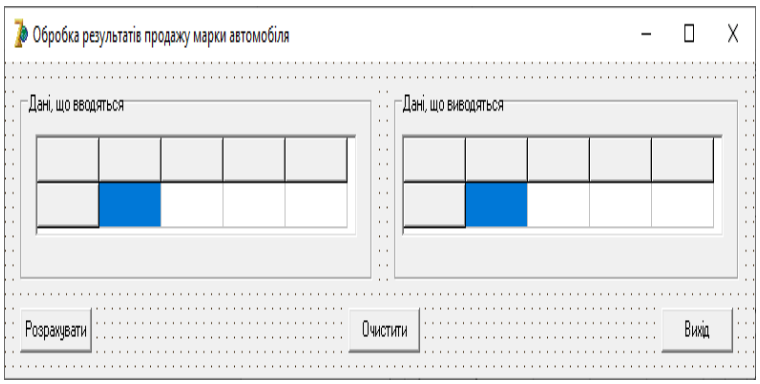


Рис. 2.8. Компоненти та вікно форми «***Form1***» (Форма1) із зміненими властивостями

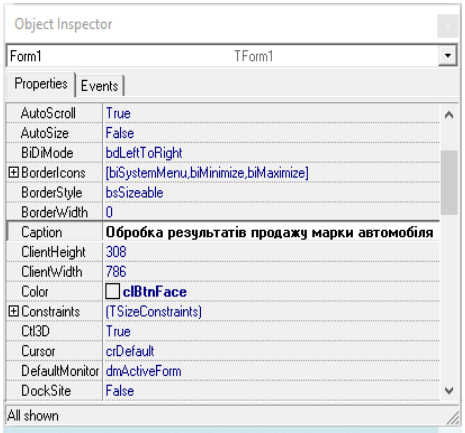


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «***Form1***» (Форма1) у вікні «***Object Inspector***» (Інспектор об'єктів) на **Обробка результатів продажу марки автомобіля**

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector*»** (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick
frmRezult	OnCreate	frmRezultCreate
GridInput	OnKeyPress	GridInputKeyPress

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);

б) у цьому ж вікні потрібно вибрати вкладку «***Events***» (Події) (рис. 2.10);

в) у лівій колонці вкладки «***Events***» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об’єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення обробника події потрібно у правій колонці вкладки «***Events***» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «***Unit_13.pas***» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «***Object Inspector***» (Інспектор об’єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «***Unit_13.pas***» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій: обробки події **OnClick** для кнопок: **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit), а також обробки події **OnCreate** для вікна форми (frmRezult) та обробки події **OnKeyPress** для полів вводу компонента (GridInput).

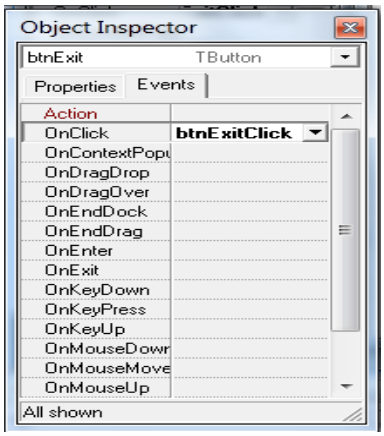


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «***Events***» (Події) у вікні «***Object Inspector***» (Інспектор об'єктів)

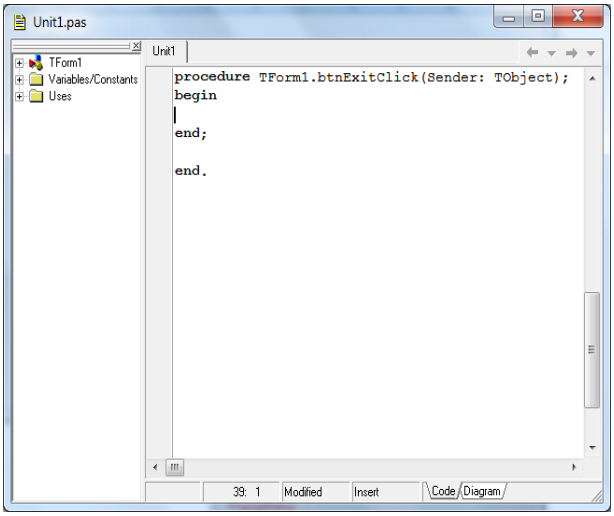


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

Лістинг програми

```
unit Unit_13;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;
type
  TfrmRezult = class(TForm)
    GroupBox1: TGroupBox;
```

```
    GroupBox2: TGroupBox;
    StrGridInput : TStringGrid;
    StrGridOutput : TStringGrid;
    btnCalculate: TButton;
    btnClear: TButton;
    btnExit: TButton;
    procedure btnCalculateClick(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure GridInputKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
end;
var
  Rezult: TRezult;
implementation
{$R *.dfm}
procedure TfrmRezult.FormCreate(Sender: TObject);
begin
   // Заголовки колонок для введення даних
   GridInput.Cells[0,0] :='Марка';
   GridInput.Cells[1,0] :='1 квартал';
   GridInput.Cells[2,0] :='2 квартал ';
   GridInput.Cells[3,0] :='3 квартал ';
   GridInput.Cells[4,0] :='4 квартал ';
   // Заголовки рядків для введення даних
   GridInput.Cells[0,1] :='ВАЗ 2106';
   // Заголовки колонок для виведення даних
   GridOutput.Cells[0,0] :='Марка';
   GridOutput.Cells[1,0] :='Сума';
   GridOutput.Cells[2,0] :='Середнє';
   GridOutput.Cells[3,0] :='Максимум';
```

```
   GridOutput.Cells[4,0] :='Мінімум';
   // Заголовки рядків для виведення даних
   GridOutput.Cells[0,1] :='ВА3 2106';
end;
procedure TfrmRezult.btnCalculateClick(Sender: TObject);
const
  // Оголошення констант
  c=4;
var
  // Оголошення масиву
  a:array[1..c] of integer;
  i : Integer;
// Оголошення змінних
  sum, max, min : Integer;
  sr : real;
begin
// Ввід даних в масив із захистом полів вводу
    for i:=1 to c do
    begin
      if GridInput.Cells[i,1]<>' ' then
        a[i]:=StrToInt(GridInput.Cells[i,1])
      else
      begin
       Application.MessageBox('У полі вводу порожньо.'#13#10'Введіть число!',
'Помилка', mb_IconStop + mb_OK);
       GridInput.Setfocus;
       exit;
      end;
    end;
  // Визначення суми та середнього
   sum := 0;
   for i:=1 to c do
   begin
     sum := sum + a[i];
     sr := sum / c;
```

```
  end;
    // Вивід даних
   GridOutput.Cells[1,1]:= IntToStr(sum);
   GridOutput.Cells[2,1]:= FloatToStr(sr);
  // Визначення мінімального та максимального
  max:=a[1];
  min:=a[1];
   for i:=1 to c do
   begin
      if a[i]<min then  min:=a[i]
   else
      if a[i]>max then  max:=a[i];
   end;
     // Вивід даних
     GridOutput.Cells[3,1]:= IntToStr(max);
     GridOutput.Cells[4,1]:= IntToStr(min);
end;
procedure TfrmRezult.GridInputKeyPress(Sender: TObject; var Key: Char);
begin
    case Key of
    '0'..'9': ; // Дозволено ввід цифр від 0 до 9
    #8: ; // Дозволено використання клавіші <Backspace>
    else
    key := #0; // Ввід інших цифр та символів заборонено
    end;
end;
procedure TfrmRezult.btnExitClick(Sender: TObject);
begin
   // Закриття програми
   Close;
end;
procedure TfrmRezult.btnClearClick(Sender: TObject);
const
  // Оголошення констант
  c=4;
```

```
var
  // Оголошення масиву
  a:array[1..c] of integer;
  i  : Integer;
begin
  // Очищення полів вводу даних
    for i:=1 to c do
    begin
      GridInput.Cells[i,1]:=' ';
    end;
  // Очищення полів виводу даних
    for i:=1 to c do
    begin
      GridOutput.Cells[i,1]:=' ';
    end;
 end;
end.
```

2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

Компіляція програми виконується з «***Головного***» меню **Project** (Проєкт) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

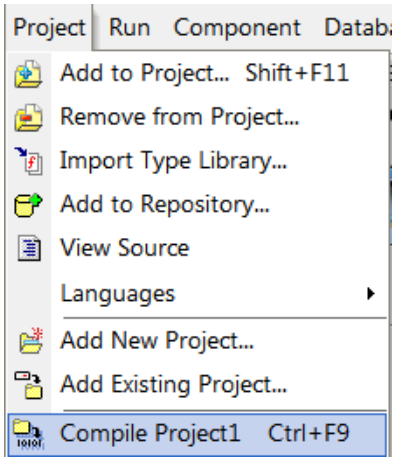


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «***Головного***» меню **Project** (Проєкт)

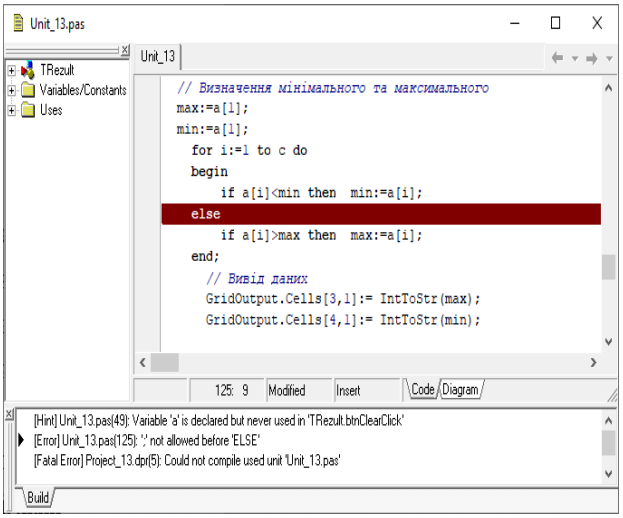


Рис. 2.13. Повідомлення компілятора про виявлені помилки

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

2.4.5. Запуск програми

Запуск програми може виконуватися з «***Головного***» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «***Debug***» (Налагодження) (рис. 2.14) або натиснути клавішу **F9.**

Рис. 2.14. Панель інструментів «***Debug***» (Налагодження)

2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами винятків є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «***Головного***» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

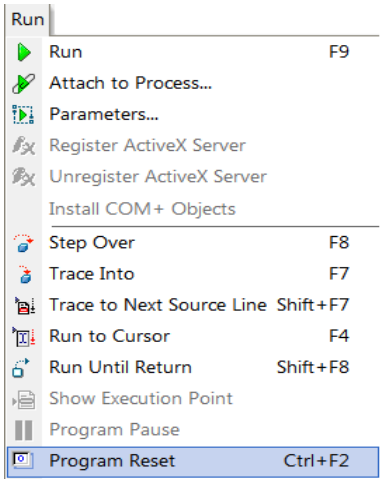


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «***Головного***» меню **Run** (Запуск)

2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №13 передбачимо такі види захисту полів вводу (див. п. п. 2.4.3):

- захист полів вводу від вводу порожнього рядка;
- захист полів вводу від вводу літер та деяких символів, використовуючи подію OnKeyPress.

3. Зміст звіту з лабораторної роботи №13

Звіт з лабораторної роботи №13 складається з:

- титульного аркуша (див. Додаток Н);
- аркуша (див. Додаток Н) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть **умову**, яка розглядається в лабораторній роботі №13, див. п. п. 2.4);
- аркуша (див. Додаток Н) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №13, див. п. п. 2.4.1);
- аркушів (див. Додаток Н) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №13, див. п. п. 2.4.3) та **висновки**.

Лабораторна робота №14

Тема: «Структури даних. Двомірний статичний масив

Мета роботи: Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп`ютерного забезпечення в технологічних процесах обробки металів різанням.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №14.
3. Скласти звіт з лабораторної роботи №14 за наведеною формою (див. Додаток О).

Хід виконання роботи:

1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

2. Алгоритм виконання лабораторної роботи №14

2.1. Запуск програми Delphi

У системному меню «***Пуск***», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7**.

2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту, виберіть в «***Головному»*** меню **File** (Файл) команду **New** > **Application** (Нова > Програма) (рис. 2.1).

2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 14 – Lab_14).

Рекомендується імені проєкту надавати ім’я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 14 – Project_14).

Файл проєкту Project _14 потрібно зберегти в папці Lab_14.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім’я, як у проєкті (наприклад, Unit _14). Його також потрібно зберегти у папці Lab_14.

На панелі інструментів «***Standard***» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

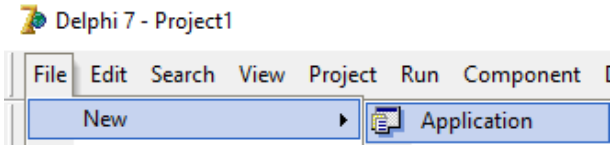


Рис. 2.1. Виберіть **New** > **Application** (Нова > Програма) в «***Головному***» меню **File** (Файл)

Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «***Standard***» (Стандартна)

2.4. Розробка програми

Розробити програму для обчислення кількість проданих за рік автомобілів різних марок, максимальну та мінімальну кількість проданих автомобілів різних марок за рік, а також скільки продали автомобілів різних марок в середньому за рік. Різні марки автомобілів продаються поквартально по-різному. Марки автомобілів та кількість проданих автомобілів поквартально прийняти самостійно.

2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рис. 2.3. Вивід повідомлення програми представлено на рис. 2.4.

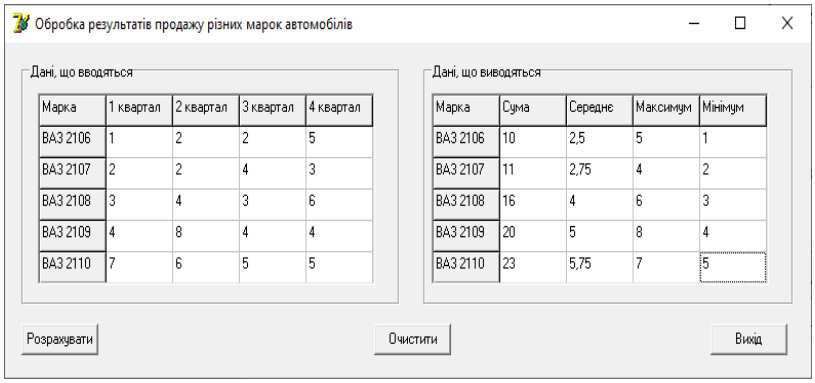


Рис. 2.3. Інтерфейс програми

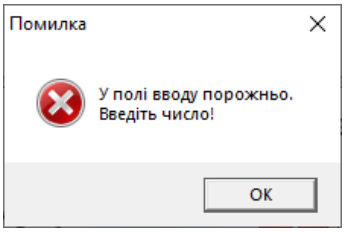


Рис. 2.4. Вивід повідомлення програми

2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «***Form1***» (Форма1) (рис. 2.5). Вікно форми «***Form1***» (Форма1) являє собою шаблон вікна розробляємої програми.

2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «***Form1***» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

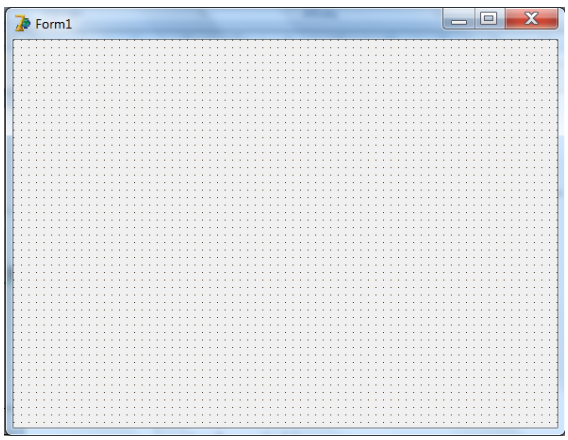


Рис. 2.5. Вікно форми «***Form1***» (Форма1)

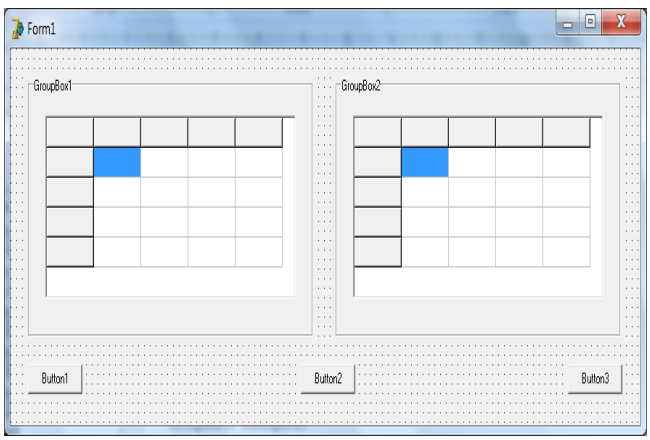


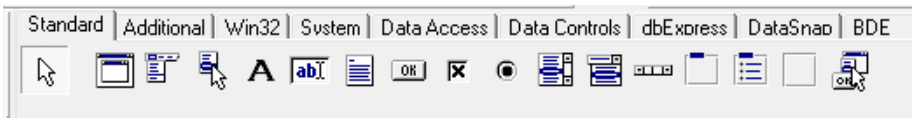
Рис. 2.6. Компоненти розміщені на вікні форми «***Form1***» (Форма1)

Для цього необхідно виконати наступні дії:

а) на «***Панелі компонентів***» на вкладці «***Standard***» (Стандартна) (рис. 2.7, а) та на вкладці «***Additional***» (Додатково) (рис. 2.7, б), вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

б) потім вказати курсором миші у довільному місці на формі «***Form1***» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «***Form1***» (Форма1) (рис. 2.6).



а

б

Рис. 2.7. «***Панель компонентів***»: а) вкладка «***Standard***» (Стандартна), б) вкладка «***Additional***» (Додатково)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	**Піктограма**	**Властивість**	**Значення**
Form1	-	Caption	Обробка результатів продажу різних марок автомобілів
		BorderStyle	bsSingle
		Name	frmRezult
Groupbox1		Caption	Дані, що вводяться
Groupbox2		Caption	Дані, що виводяться
StringGrid1		Name	GridInput
		ColCount	5
		RowCount	6
		FixedCols	1
		FixedRows	1
		Options:	Розкрити список, натиснувши на знак «+»
		goEditing	True
		goTabs	True
		goAlwaysShowEditing	True

Продовження таблиці 2.1

Компонент	Піктограма	Властивість	Значення
StringGrid2		Name	GridOutput
		ColCount	5
		RowCount	6
		FixedCols	1
		FixedRows	1
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «***Form1***» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1. Увага! Найменування колонок та рядків компонентів StringGrid1 та StringGrid2 можливе лише у вихідному коді.

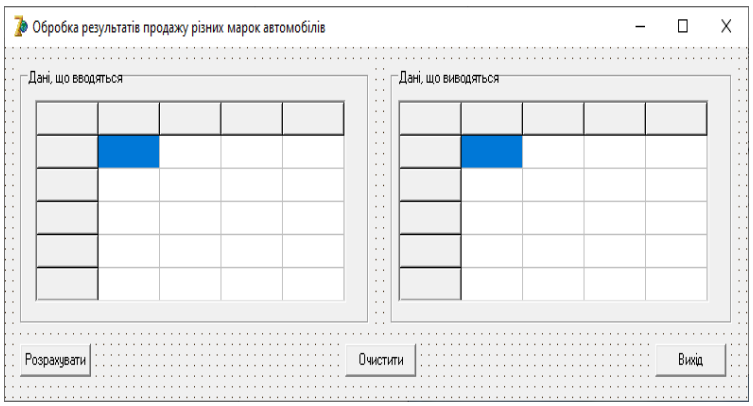


Рис. 2.8. Компоненти та вікно форми «***Form1***» (Форма1) із зміненими властивостями

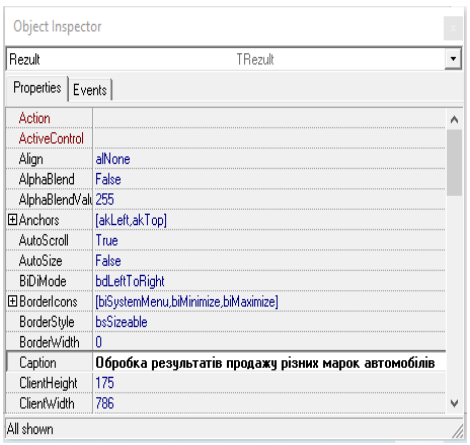


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «***Form1***» (Форма1) у вікні «***Object Inspector***» (Інспектор об'єктів) на **Обробка результатів продажу різних марок автомобілів**

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick
frmRezult	OnCreate	frmRezultCreate
GridInput	OnKeyPress	GridInputKeyPress

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);

б) у цьому ж вікні потрібно вибрати вкладку «***Events***» (Події) (рис. 2.10);

в) у лівій колонці вкладки «***Events***» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об’єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення обробника події потрібно у правій колонці вкладки «***Events***» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «***Unit_14.pas***» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «***Object Inspector***» (Інспектор об’єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «***Unit_14.pas***» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій: обробки події **OnClick** для кнопок: **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit), а також обробки події **OnCreate** для вікна форми (frmRezult) та обробки події **OnKeyPress** для компонента введення даних (GridInput).

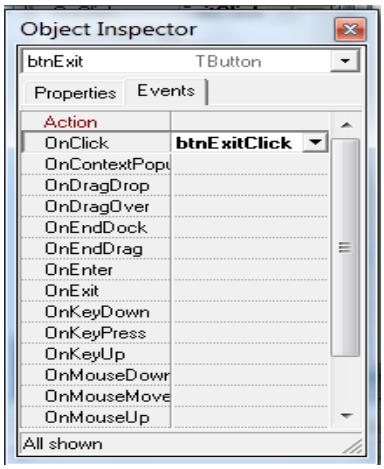


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «***Events***» (Події) у вікні «***Object Inspector***» (Інспектор об'єктів)

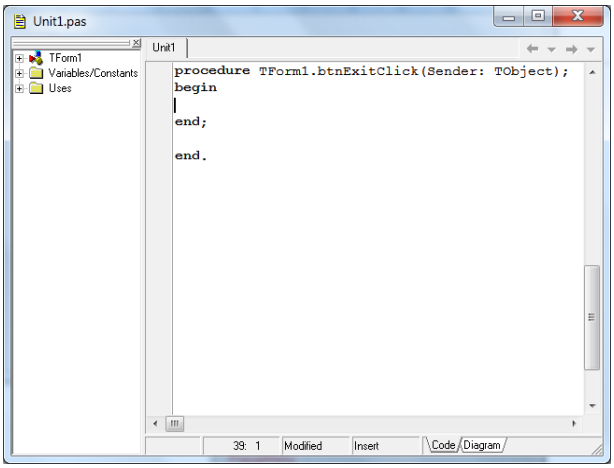


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

Лістинг програми

```
unit Unit_14;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;
type
  TfrmRezult = class(TForm)
```

```
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    StrGridInput : TStringGrid;
    StrGridOutput : TStringGrid;
    btnCalculate: TButton;
    btnClear: TButton;
    btnExit: TButton;
    procedure btnCalculateClick(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure GridInputKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
end;
var
  Rezult: TRezult;
implementation
{$R *.dfm}
procedure TfrmRezult.FormCreate(Sender: TObject);
begin
   // Заголовки колонок для введення даних
   GridInput.Cells[0,0] :='Марка';
   GridInput.Cells[1,0] :='1 квартал';
   GridInput.Cells[2,0] :='2 квартал ';
   GridInput.Cells[3,0] :='3 квартал ';
   GridInput.Cells[4,0] :='4 квартал ';
   // Заголовки рядків для введення даних
   GridInput.Cells[0,1] :='ВАЗ 2106';
   GridInput.Cells[0,2] :='ВАЗ 2107';
   GridInput.Cells[0,3] :='ВАЗ 2108';
   GridInput.Cells[0,4] :='ВАЗ 2109';
   GridInput.Cells[0,5] :='ВАЗ 2110';
```

```
  // Заголовки колонок для виведення даних
  GridOutput.Cells[0,0] :='Марка';
  GridOutput.Cells[1,0] :='Сумма';
  GridOutput.Cells[2,0] :='Среднее';
  GridOutput.Cells[3,0] :='Максимум';
  GridOutput.Cells[4,0] :='Минимум';
  // Заголовки рядків для виведення даних
  GridOutput.Cells[0,1] :='ВАЗ 2106';
  GridOutput.Cells[0,2] :='ВАЗ 2107';
  GridOutput.Cells[0,3] :='ВАЗ 2108';
  GridOutput.Cells[0,4] :='ВАЗ 2109';
  GridOutput.Cells[0,5] :='ВАЗ 2110';
end;
procedure TfrmRezult.btnCalculateClick(Sender: TObject);
const
 // Оголошення констант
 c=4; r=5;
var
 // Оголошення масиву
 a : array[1..c, 1..r] of integer;
 i, j : Integer;
// Оголошення змінних
 sum, max, min : Integer;
 sr : real;
begin
// Ввід даних в масив із захистом полів вводу
 for j:=1 to r do
 begin
  for i:=1 to c do
  begin
    if GridInput.Cells[i,j]<>' ' then
      a[i,j]:=StrToInt(GridInput.Cells[i,j])
    else
    begin
     Application.MessageBox('У полі вводу порожньо.'#13#10'Введіть число!',
```

```
      'Помилка', mb_IconStop + mb_OK);
      GridInput.Setfocus;
      exit;
     end;
  end;
end;
// Визначення суми та середнього
for j:=1 to r do
begin
  sum := 0;
  for i:=1 to c do
  begin
    sum := sum + a[i,j];
    sr := sum / c;
  end;
   // Вивід даних
  GridOutput.Cells[1,j]:= IntToStr(sum);
  GridOutput.Cells[2,j]:= FloatToStr(sr);
end;
// Визначення мінімального та максимального
for j:=1 to r do
begin
   max:=a[1,j] ;
   min:=a[1,j];
   for i:=1 to c do
   begin
     if a[i,j]<min then
     begin
       min:=a[i,j];
     end
     else
     if a[i,j]>max then
     begin
       max:=a[i,j];
     end;
```

```
      end;
      // Вивід даних
      GridOutput.Cells[3,j]:= IntToStr(max);
      GridOutput.Cells[4,j]:= IntToStr(min);
  end;
end;
procedure TfrmRezult.GridInputKeyPress(Sender: TObject; var Key: Char);
begin
     case Key of
     '0'..'9': ; // Дозволено ввід цифр від 0 до 9
     #8: ; // Дозволено використання клавіші <Backspace>
     else
     key := #0; // Ввід інших цифр та символів заборонено
     end;
end;
procedure TfrmRezult.btnExitClick(Sender: TObject);
begin
   // Закриття програми
   Close;
end;
procedure TfrmRezult.btnClearClick(Sender: TObject);
const
  // Оголошення констант
  c=4; r=5;
var
  // Оголошення масиву і змінних
  a : array[1..c, 1..r] of integer;
  i, j : Integer;
begin
  // Очистка полів вводу даних
  for j:=1 to r do
  begin
    for i:=1 to c do
    begin
      GridInput.Cells[i,j]:=' ';
```

```
    end;
  end;
  // Очищення полів виводу даних
  for j:=1 to r do
  begin
    for i:=1 to c do
    begin
      GridOutput.Cells[i,j]:=' ';
    end;
  end;
end;
end.
```

2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

Компіляція програми виконується з «***Головного***» меню **Project** (Проєкт) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

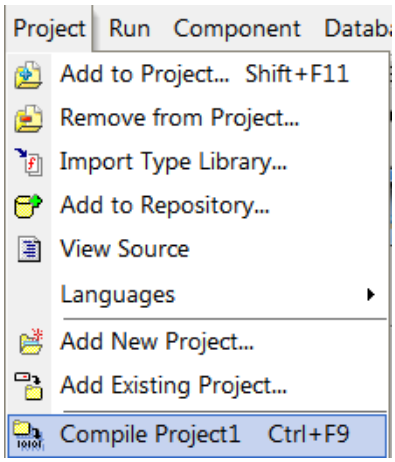


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «***Головного***» меню **Project** (Проєкт)

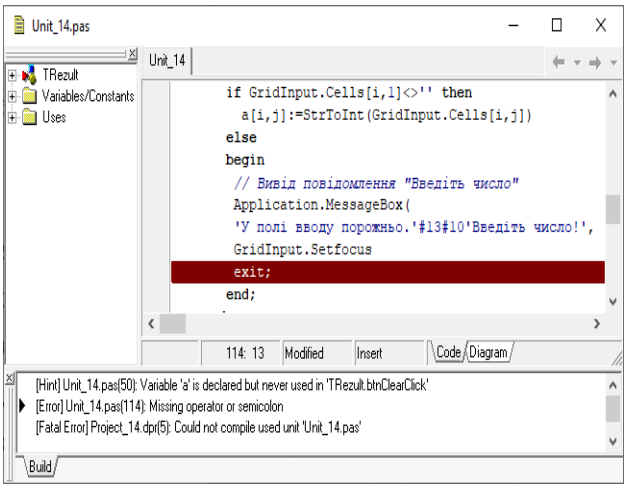


Рис. 2.13. Повідомлення компілятора про виявлені помилки

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

2.4.5. Запуск програми

Запуск програми може виконуватися з «***Головного***» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «***Debug***» (Налагодження) (рис. 2.14) або натиснути клавішу **F9.**

Рис. 2.14. Панель інструментів «***Debug***» (Налагодження)

2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами винятків є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «***Головного***» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

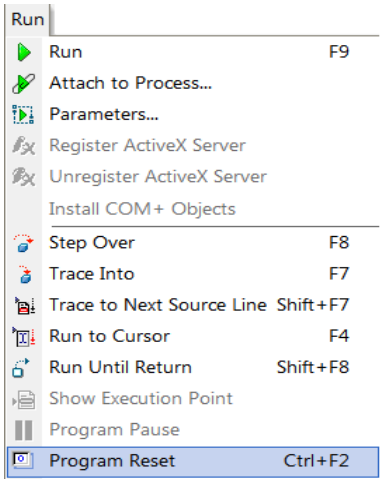


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «***Головного***» меню **Run** (Запуск)

2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №14 передбачимо такі види захисту полів вводу (див. п. п. 2.4.3):

- захист полів вводу від вводу порожнього рядка;

- захист полів вводу від вводу літер та деяких символів, використовуючи подію OnKeyPress.

3. Зміст звіту з лабораторної роботи №14

Звіт з лабораторної роботи №14 складається з:

- титульного аркуша (див. Додаток О);

- аркуша (див. Додаток О) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть **умову**, яка розглядається в лабораторній роботі №14, див. п. п. 2.4);

- аркуша (див. Додаток О) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №14, див. п. п. 2.4.1);

- аркушів (див. Додаток О) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №14, див. п. п. 2.4.3) та **висновки**.

Лабораторна робота №15

Тема: «Структури даних. Одномірний динамічний масив

Мета роботи: Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп`ютерного забезпечення в технологічних процесах обробки металів різанням.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №15.
3. Скласти звіт з лабораторної роботи №15 за наведеною формою (див. Додаток П).

Хід виконання роботи:

1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

2. Алгоритм виконання лабораторної роботи №15

2.1. Запуск програми Delphi

У системному меню «***Пуск***», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7**.

2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту, виберіть в «***Головному»*** меню **File** (Файл) команду **New** > **Application** (Нова > Програма) (рис. 2.1).

2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 15 – Lab_15).

Рекомендується імені проєкту надавати ім’я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 15 – Project_15).

Файл проєкту Project _15 потрібно зберегти в папці Lab_15.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім’я, як у проєкті (наприклад, Unit _15). Його також потрібно зберегти у папці Lab_15.

На панелі інструментів «***Standard***» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

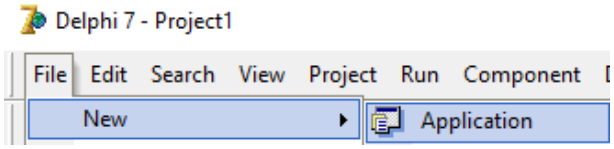


Рис. 2.1. Виберіть **New** > **Application** (Нова > Програма) в «***Головному***» меню **File** (Файл)

Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «***Standard***» (Стандартна)

2.4. Розробка програми

Розробити програму для обчислення кількість проданих за рік автомобілів однієї марки, максимальну та мінімальну кількість проданих автомобілів однієї марки за рік, а також скільки продали автомобілів однієї марки в середньому за рік. Одна марка автомобіля продається поквартально по-різному. Марку автомобіля та кількість проданих автомобілів поквартально прийняти самостійно.

2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рис. 2.3 (а, б, в, г). Вивід повідомлення програми представлено на рис. 2.4 (а, б).

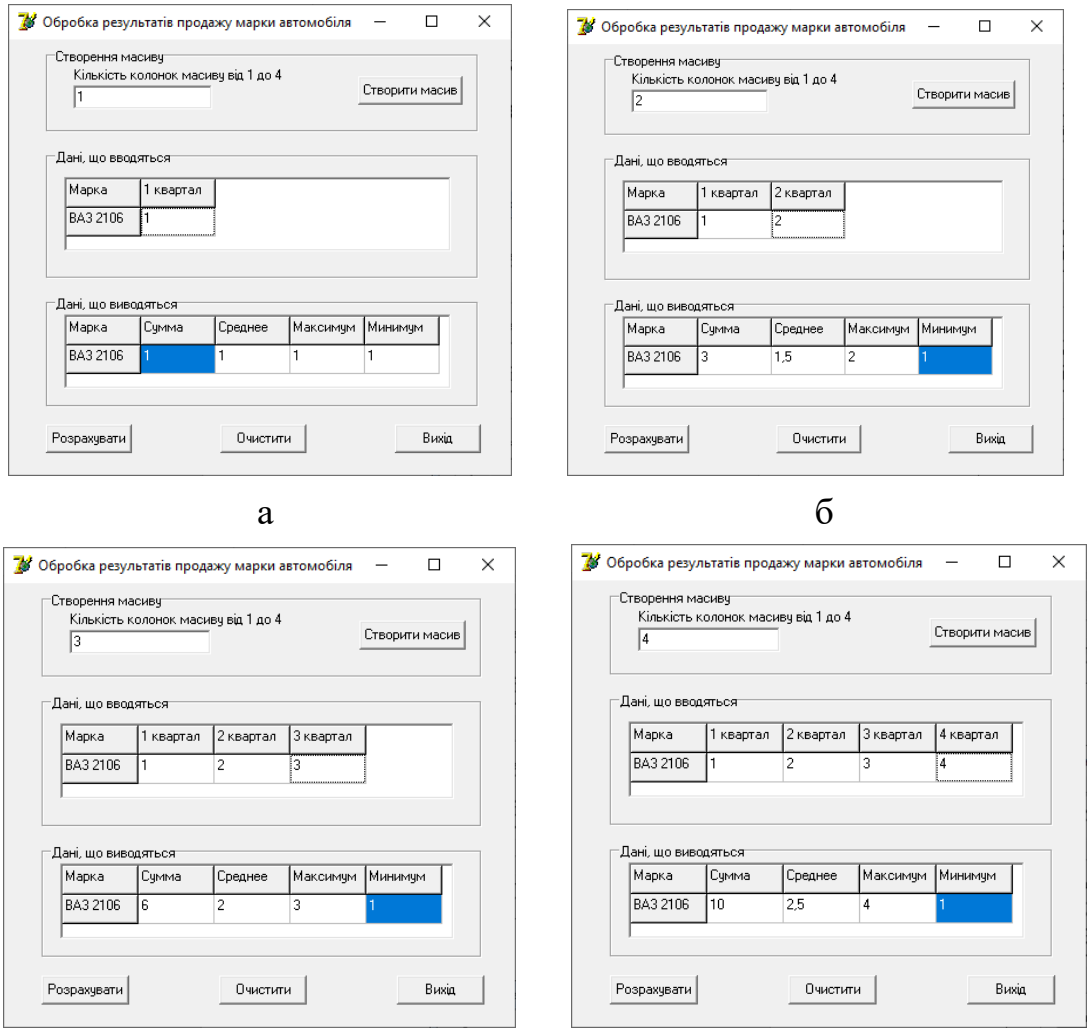


Рис. 2.3. Інтерфейс програми: а, б, в, г

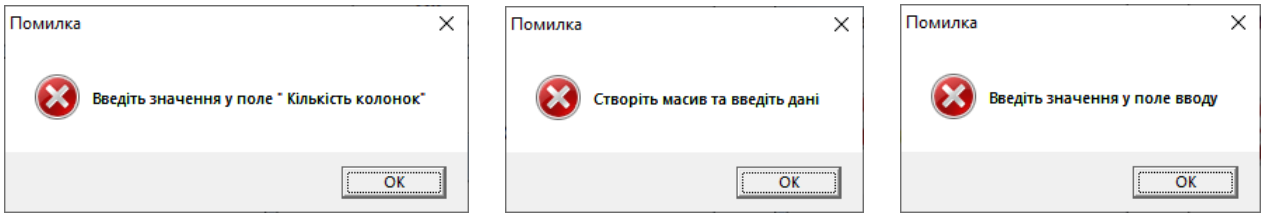


Рис. 2.4. Вивід повідомлення програми: а, б, в

2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «***Form1***» (Форма1) (рис. 2.5). Вікно форми «***Form1***» (Форма1) являє собою шаблон вікна розробляємої програми.

2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «***Form1***» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

Для цього необхідно виконати наступні дії:

а) на «***Панелі компонентів***» на вкладці «***Standard***» (Стандартна) (рис. 2.7, а) та на вкладці «***Additional***» (Додатково) (рис. 2.7, б), вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

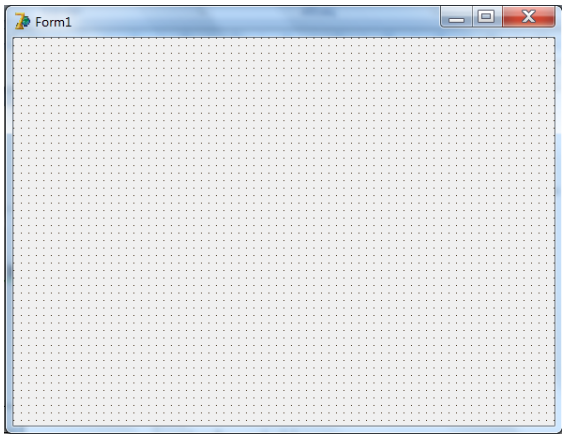


Рис. 2.5. Вікно форми «***Form1***» (Форма1)

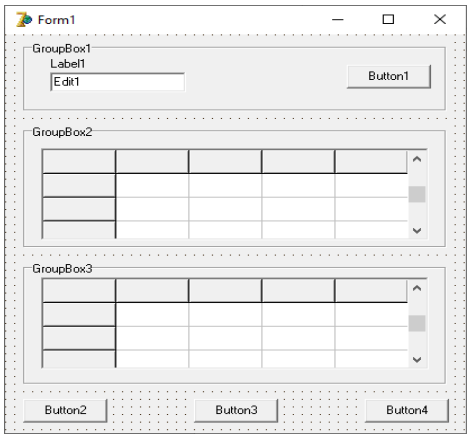
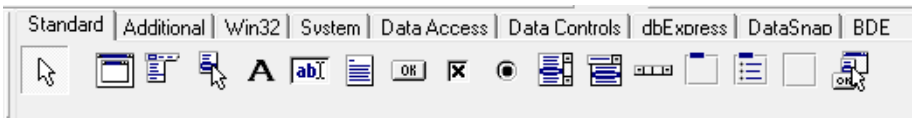


Рис. 2.6. Компоненти розміщені на вікні форми «***Form1***» (Форма1)



а

б

Рис. 2.7. «***Панель компонентів***»: а) вкладка «***Standard***» (Стандартна), б) вкладка «***Additional***» (Додатково)

б) потім вказати курсором миші у довільному місці на формі «***Form1***» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «***Form1***» (Форма1) (рис. 2.6).

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	**Піктограма**	**Властивість**	**Значення**
Form1	-	Caption	Обробка результатів продажу марки автомобіля
		BorderStyle	bsSingle
		Name	frmRezult
Groupbox1		Caption	Створення масиву
Groupbox2		Caption	Дані, що вводяться
Groupbox3		Caption	Дані, що виводяться
Label1		Caption	Кількість колонок масиву від 1 до 4
Edit1		Name	edtInputn
		Text	значення не вводимо, а тільки очистити поле
StringGrid1		Name	GridInput
		ColCount	2
		RowCount	2
		FixedCols	1
		FixedRows	1
StringGrid2		Name	GridOutput
		ColCount	5
		RowCount	2
		FixedCols	1
		FixedRows	1
Button1		Name	btnCreate
		Caption	Створити масив
Button2		Name	btnCalculate
		Caption	Розрахувати
Button3		Name	btnClear
		Caption	Очистити
Button4		Name	btnExit
		Caption	Вихід

2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «***Form1***» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1. Увага! Найменування колонок та рядків компонентів StringGrid1 та StringGrid2 можливе лише у вихідному коді.

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

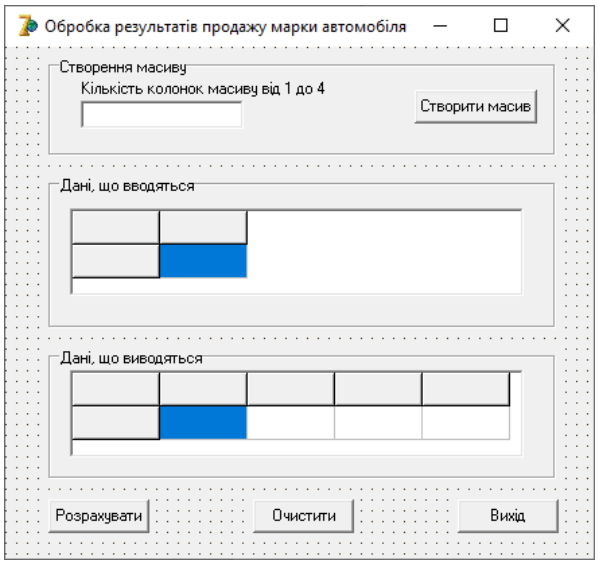


Рис. 2.8. Компоненти та вікно форми «***Form1***» (Форма1) із зміненими властивостями

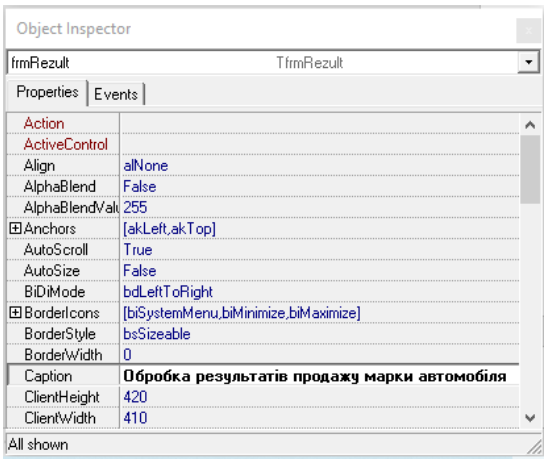


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «***Form1***» (Форма1) у вікні «***Object Inspector***» (Інспектор об'єктів) на **Обробка результатів продажу марки автомобіля**

2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCreate	OnClick	btnCreateClick
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick
frmRezult	OnCreate	frmRezultCreate
edtInputn	OnKeyPress	edtInputnKeyPress
GridInput	OnKeyPress	GridInputKeyPress

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);

б) у цьому ж вікні потрібно вибрати вкладку «***Events***» (Події) (рис. 2.10);

в) у лівій колонці вкладки «***Events***» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об’єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення функції обробника події потрібно у правій колонці вкладки «***Events***» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «***Unit_15.pas***» (Редактор коду), до якого буде додано шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «***Object Inspector***» (Інспектор об’єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «***Unit_15.pas***» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п. 2.4.3, наведені тексти інструкцій: обробки події **OnClick** для кнопок: **Створення масиву** (btnCreate), **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit), а також обробки події **OnCreate** для вікна форми (frmRezult) та обробки події **OnKeyPress** для полів вводу компонентів (GridInput) і (edtInputn).

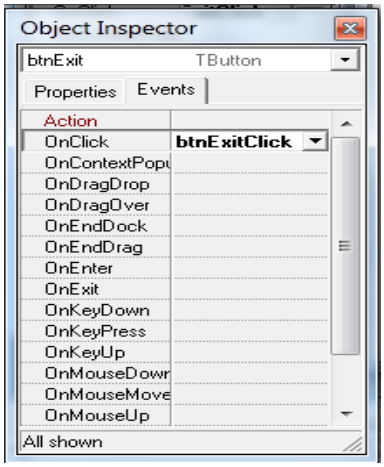


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «***Events***» (Події) у вікні «***Object Inspector***» (Інспектор об'єктів)

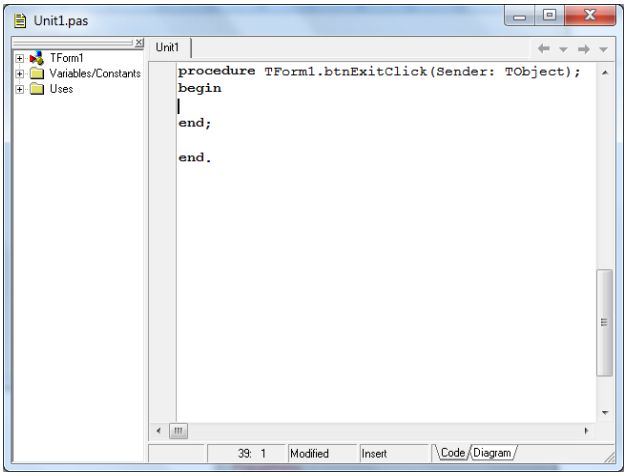


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

Лістинг програми

```
unit Unit_15;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;
type
  TfrmRezult = class(TForm)
    btnCreate: TButton;
    GridInput: TStringGrid;
    edtInputn: TEdit;
    btnCalculate: TButton;
    btnClear: TButton;
    btnExit: TButton;
    GridOutput: TStringGrid;
```

```
    Label1: TLabel;
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    procedure btnCreateClick(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure btnExitClick(Sender: TObject);
    procedure btnCalculateClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure edtInputnKeyPress(Sender: TObject; var Key: Char);
    procedure GridInputKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
end;
var
  frmRezult: TfrmRezult;
  // Оголошення масиву
  a:array of integer;
  // Оголошення змінних
  i:integer; n:integer;
implementation
{$R *.dfm}
procedure TfrmRezult.btnCreateClick(Sender: TObject);
begin
   // Захист поля вводу n
   if edtInputn.text=' ' then
   begin
   Application.MessageBox('Введіть значення у поле " Кількість колонок"',
   'Помилка',mb_ok+mb_IconError);
   edtInputn.SetFocus;
   exit;
   end;
    // Завдання кількості колонок для масиву
```

```
    n:=strtoint(edtInputn.text);
    // Встановлення кількості колонок для масиву
    SetLength(a,n);
    // Встановлення кроку колонок в компоненте
    GridInput.ColCount:=n+1;
    // Встановлення опцій редагування
GridInput.Options:=[goFixedVertLine,goFixedHorzLine,goVertLine,goHorzLine,go
RangeSelect,goEditing,goTabs];
    // Заголовки рядків для введення даних
    GridInput.Cells[0,1]:='ВА3 2106';
    // Заголовки колонок для введення даних
    GridInput.Cells[0,0]:='Марка';
    GridInput.Cells[1,0]:='1 квартал';
    GridInput.Cells[2,0]:='2 квартал';
    GridInput.Cells[3,0]:='3 квартал';
    GridInput.Cells[4,0]:='4 квартал';
end;
procedure TfrmRezult.FormCreate(Sender: TObject);
begin
   // Заголовки колонок для виведення даних
   GridOutput.Cells[0,0] :='Марка';
   GridOutput.Cells[1,0] :='Сумма';
   GridOutput.Cells[2,0] :='Среднее';
   GridOutput.Cells[3,0] :='Максимум';
   GridOutput.Cells[4,0] :='Минимум';
   // Заголовки рядків для виведення даних
   GridOutput.Cells[0,1] :='ВА3 2106';
end;
procedure TfrmRezult.btnCalculateClick(Sender: TObject);
var
  // Оголошення змінних
  i : Integer;
  sum, max, min : Integer;
  sr : real;
begin
```

```
  // Перевірка створення масиву
  if n=0 then
  begin
  Application.MessageBox('Створіть масив та введіть дані',
  'Помилка',mb_ok+mb_IconError);
  edtInputn.SetFocus;
  exit;
  end;
   // Ввід даних в масив із захистом полів вводу
   for i:=0 to n-1 do
   begin
      if GridInput.Cells[i+1,1]<>' ' then
        a[i]:=StrToInt(GridInput.Cells[i+1,1])
      else
      begin
       Application.MessageBox('Введіть значення у поле вводу', 'Помилка',
mb_IconStop + mb_OK);
       GridInput.SetFocus;
       exit;
      end;
    end;
  // Визначення суми та середнього
   sum := 0;
   for i:=0 to n-1 do
   begin
      sum := sum + a[i];
      sr := sum / n;
   end;
     // Вивід даних
    GridOutput.Cells[1,1]:= IntToStr(sum);
    GridOutput.Cells[2,1]:= FloatToStr(sr);
  // Визначення мінімального та максимального
  max:=a[0];
  min:=a[0];
    for i:=0 to n-1 do
```

```
   begin
      if a[i]<min then  min:=a[i]
   else
      if a[i]>max then  max:=a[i];
   end;
    // Вивід даних
    GridOutput.Cells[3,1]:= IntToStr(max);
    GridOutput.Cells[4,1]:= IntToStr(min);
end;
procedure TfrmRezult.edtInputnKeyPress(SendfrmRezultObject; var Key: Char);
begin
     case key of
    // Дозволено ввід цифр від 1до 4
   '1'..'4': begin
            // Заборонено ввід другої цифри «1»
          if pos('1',edtInputn.Text)<>0 then key:=#0; // Ввід інших цифр
        заборонено
          // Заборонено ввід другої цифри «2»
          if pos('2',edtInputn.Text)<>0 then key:=#0; // Ввід інших цифр
        заборонено
          // Заборонено ввід другої цифри «3»
          if pos('3',edtInputn.Text)<>0 then key:=#0; // Ввід інших цифр
        заборонено
          // Заборонено ввід другої цифри «4»
          if pos('4',edtInputn.Text)<>0 then key:=#0; // Ввід інших цифр
        заборонено
         end;
   #8: ; // Дозволено використання клавіши Backspace #8
   else
     key:=#0; // Ввід інших цифр та символів заборонено
   end;
end;
procedure TfrmRezult.GridInputKeyPress(Sender: TObject; var Key: Char);
begin
    case Key of
```

```
    '0'..'9': ;  // Дозволено ввід цифр від 0 до 9
    #8: ;  // Дозволено використання клавіши <Backspace>
    else
    key := #0; // Ввід інших цифр та символів заборонено
    end;
end;
procedure TfrmRezult.btnExitClick(Sender: TObject);
begin
  // Закриття програми
  Close;
  // Звільняємо пам'ять
  a:=nil;
end;
procedure TfrmRezult.btnClearClick(Sender: TObject);
var
// Оголошення змінної
 i:integer;
begin
 // Очищення полів вводу даних
   for i:=0 to n-1 do
   begin
     GridInput.Cells[i+1,1]:=' ';
   end;
 // Очищення полів виводу даних
  GridOutput.Cells[1,1]:=' ';
  GridOutput.Cells[2,1]:=' ';
  GridOutput.Cells[3,1]:=' ';
  GridOutput.Cells[4,1]:=' ';
// Очищення поля вводу кількості колонок
  edtInputn.text:= ' ';
end;
end.
```

2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому — генерується виконувана програма (exe-файл).

Компіляція програми виконується з «***Головного***» меню **Project** (Проєкт) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

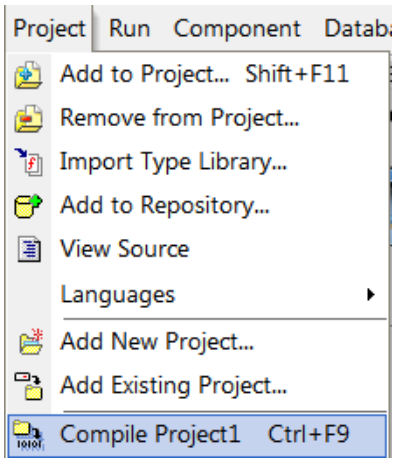


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «***Головного***» меню **Project** (Проєкт)

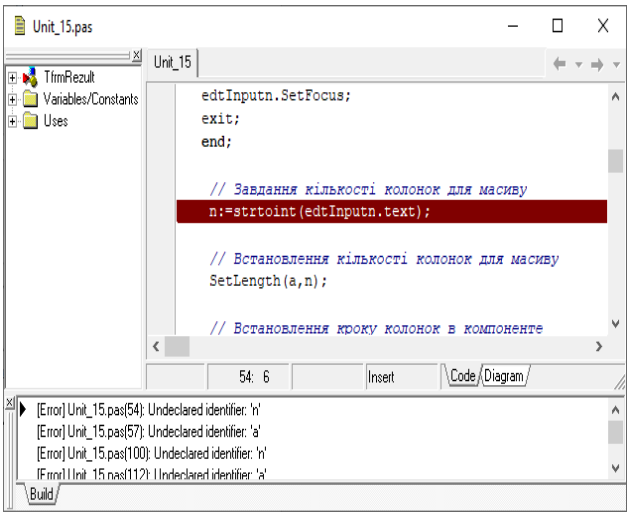


Рис. 2.13. Повідомлення компілятора про виявлені помилки

2.4.5. Запуск програми

Запуск програми може виконуватися з «***Головного***» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «***Debug***» (Налагодження) (рис. 2.14) або натиснути клавішу **F9.**

Рис. 2.14. Панель інструментів «***Debug***» (Налагодження)

2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами винятків є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «***Головного***» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

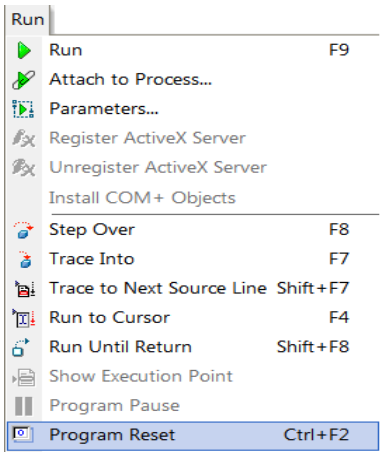


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «***Головного***» меню **Run** (Запуск)

2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №15 передбачимо такі види захисту полів вводу (див. п. п. 2.4.3):

- захист полів вводу від вводу порожнього рядка;
- захист полів вводу від вводу літер та деяких символів, використовуючи подію OnKeyPress.

3. Зміст звіту з лабораторної роботи №15

Звіт з лабораторної роботи №15 складається з:

- титульного аркуша (див. Додаток П);
- аркуша (див. Додаток П) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть **умову**, яка розглядається в лабораторній роботі №15, див. п. п. 2.4);
- аркуша (див. Додаток П) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №15, див. п. п. 2.4.1);
- аркушів (див. Додаток П) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №15, див. п. п. 2.4.3) та **висновки**.

Лабораторна робота №16

Тема: «Структури даних. Тип даних – запис»

Мета роботи: Засвоєння початкових знань та придбання навичок, необхідних для розробки програмних продуктів комп`ютерного забезпечення в технологічних процесах обробки металів різанням.

Порядок виконання роботи

1. Вивчити теоретичні відомості.
2. Виконати алгоритм виконання лабораторної роботи №16.
3. Скласти звіт з лабораторної роботи №16 за наведеною формою (див. Додаток Р).

Хід виконання роботи:

1. Теоретичні відомості

Вивчити теоретичні відомості зі списку наведеної літератури.

2. Алгоритм виконання лабораторної роботи №16

2.1. Запуск програми Delphi

У системному меню «***Пуск***», операційної системи Windows, вибрати: **Пуск > Програми > Borland Delphi 7 > Delphi 7**.

2.2. Створення проєкту

Після запуску програми Delphi створюється за замовчуванням проєкт типу – **Application** (Програма).

Delphi дозволяє розробляти різні програми, які в Delphi називаються проєктами. У лабораторних роботах створюються проєкти – **Application** (Програма), під операційну систему Windows з графічним інтерфейсом.

Для створення нового проєкту, виберіть в «***Головному»*** меню **File** (Файл) команду **New** > **Application** (Нова > Програма) (рис. 2.1).

2.3. Збереження проєкту

Delphi для кожного проєкту створює декілька файлів. Щоб файли різних проєктів не переплуталися між собою, слід для кожного проєкту створювати свою **папку** з іменем, яке містить назву та номер лабораторної роботи (наприклад, лабораторна робота 16 – Lab_16).

Рекомендується імені проєкту надавати ім'я, яке містить номер лабораторної роботи (наприклад, проєкт для лабораторної роботи 16 – Project_16).

Файл проєкту Project _16 потрібно зберегти в папці Lab_16.

Необхідно також зберегти **файл модуля вихідного коду**.

Рекомендується імені файлу модуля вихідного коду надавати ім'я, як у проєкті (наприклад, Unit _16). Його також потрібно зберегти у папці Lab_16.

На панелі інструментів «***Standard***» (Стандартна) натиснути кнопку **Save all** (Зберегти все) та зберегти **файли проєкту** та **файл модуля вихідного коду** згідно з вище сказаними рекомендаціями (рис. 2.2).

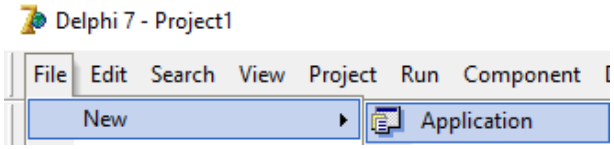


Рис. 2.1. Виберіть **New** > **Application** (Нова > Програма) в «***Головному***» меню **File** (Файл)

Рис. 2.2. Натиснути кнопку **Save all** (Зберегти все) на панелі інструментів «***Standard***» (Стандартна)

2.4. Розробка програми

Розробити програму для обчислення середньої зарплати у кожного робітника на ділянці за 3 місяці. Кількість робітників прийняти у кількості 5 осіб. Визначити середню зарплату робочих дільниць за 3 місяці.

2.4.1. Створення інтерфейсу програми

Інтерфейс програми представлений на рис. 2.3. Вивід повідомлення програми представлено на рис. 2.4.

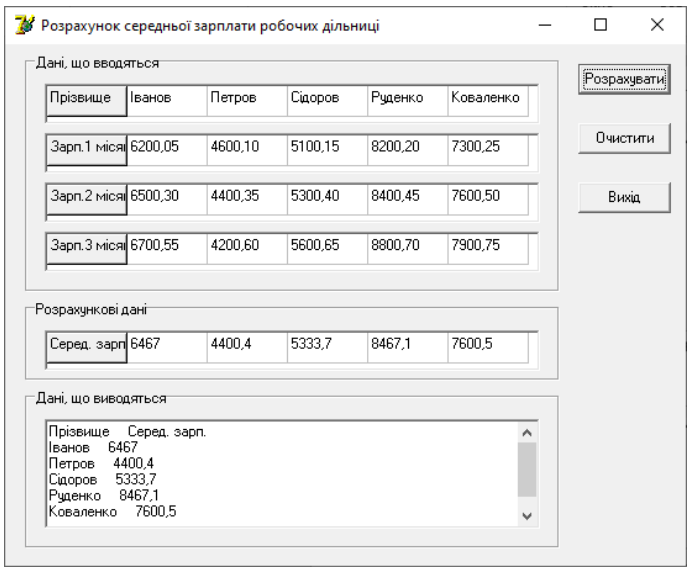


Рис. 2.3. Інтерфейс програми

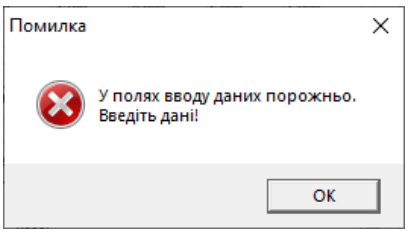


Рис. 2.4. Вивід повідомлення програми

2.4.1.1. Вікно форми Form1 (Форма1)

При створенні проєкту типу – **Application** (Програма), Delphi автоматично створює вікно форми «***Form1***» (Форма1) (рис. 2.5). Вікно форми «***Form1***» (Форма1) являє собою шаблон вікна розробляємої програми.

2.4.1.2. Розміщення компонентів на вікні форми Form1 (Форма1)

На вікні форми «***Form1***» (Форма1) розмістити компоненти (рис. 2.6) вказані в таблиці 2.1.

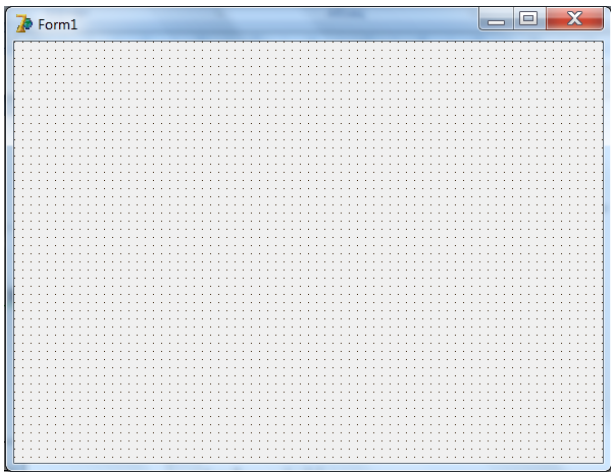


Рис. 2.5. Вікно форми «***Form1***» (Форма1)

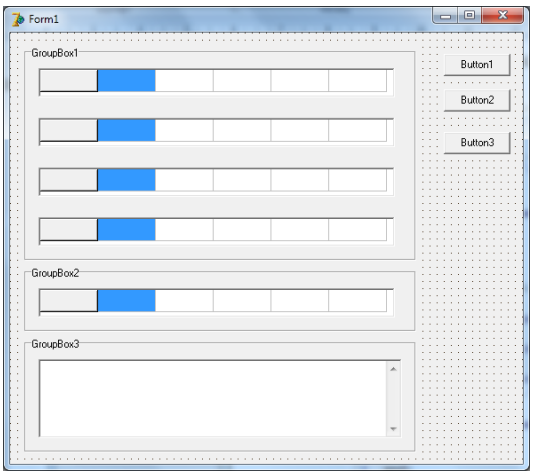


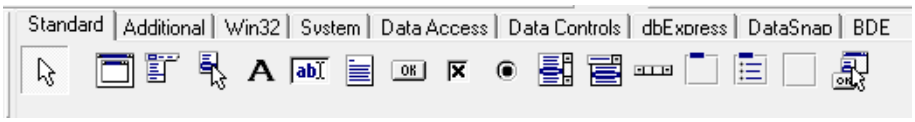
Рис. 2.6. Компоненти розміщені на вікні форми «***Form1***» (Форма1)

Для цього необхідно виконати наступні дії:

а) на «***Панелі компонентів***» на вкладці «***Standard***» (Стандартна) (рис. 2.7, а) та на вкладці «***Additional***» (Додатково) (рис. 2.7, б), вказати курсором миші на піктограму потрібного компонента (табл. 2.1);

б) потім вказати курсором миші у довільному місці на формі «***Form1***» (Форма1) (рис. 2.6);

в) якщо необхідно, перетягніть компонент у потрібне місце на формі «***Form1***» (Форма1) (рис. 2.6).



а

б

Рис. 2.7. «***Панель компонентів***»: а) вкладка «***Standard***» (Стандартна), б) вкладка «***Additional***» (Додатково)

Таблиця 2.1 – Компоненти та їх властивості для інтерфейсу програми

Компонент	**Піктограма**	**Властивість**	**Значення**
Form1	-	Caption	Розрахунок середньої зарплати робочих дільниці
		BorderStyle	bsSingle
		Name	frmRezult
Groupbox1		Caption	Дані, що вводяться
Groupbox2		Caption	Розрахункові дані
Groupbox3		Caption	Дані, що виводяться
StringGrid1		Name	GridInputfam
		ColCount	6
		RowCount	1
		FixedCols	1
		FixedRows	0
		Options:	Розкрити список, натиснувши на знак «+»
		goEditing	True
		goTabs	True
		goAlwaysShowEditing	True

Продовження таблиці 2.1

Компонент	Піктограма	Властивість	Значення
StringGrid2		Name	GridInputz1
		ColCount	6
		RowCount	1
		FixedCols	1
		FixedRows	0
		Options:	Розкрити список, натиснувши на знак «+»
		goEditing	True
		goTabs	True
		goAlwaysShowEditing	True
StringGrid3		Name	GridInputz2
		ColCount	6
		RowCount	1
		FixedCols	1
		FixedRows	0
		Options:	Розкрити список, натиснувши на знак «+»
		goEditing	True
		goTabs	True
		goAlwaysShowEditing	True
StringGrid4		Name	GridInputz3
		ColCount	6
		RowCount	1
		FixedCols	1
		FixedRows	0
		Options:	Розкрити список, натиснувши на знак «+»
		goEditing	True
		goTabs	True
		goAlwaysShowEditing	True
StringGrid5		Name	GridOutputSredz
		ColCount	6
		RowCount	1
		FixedCols	1
		FixedRows	0

Продовження таблиці 2.1

Компонент	Піктограма	Властивість	Значення
Memo1		Name	memOutput
		ScrollBars	ssVertical
		ReadOnly	True
		Lines	натиснути кнопку ... У з'явившемуся вікні очистити поле, потім натиснути кнопку **ОК**
Button1		Name	btnCalculate
		Caption	Розрахувати
Button2		Name	btnClear
		Caption	Очистити
Button3		Name	btnExit
		Caption	Вихід

2.4.1.3. Зміна властивостей вікна форми Form1 (Форма1) та компонентів

Змінимо властивості вікна форми «***Form1***» (Форма1) та компонентів (рис. 2.8) згідно з вказаними властивостями в таблиці 2.1. Увага! Найменування рядків компонентів StringGrid1, StringGrid2, StringGrid3, StringGrid4 та StringGrid5 можливе лише у вихідному коді.

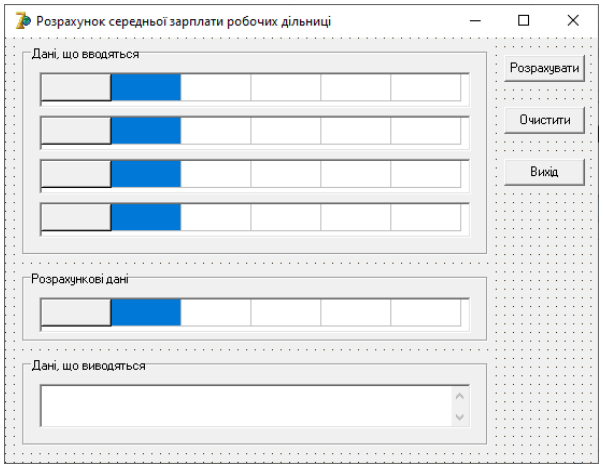


Рис. 2.8. Компоненти та вікно форми «***Form1***» (Форма1) із зміненими властивостями

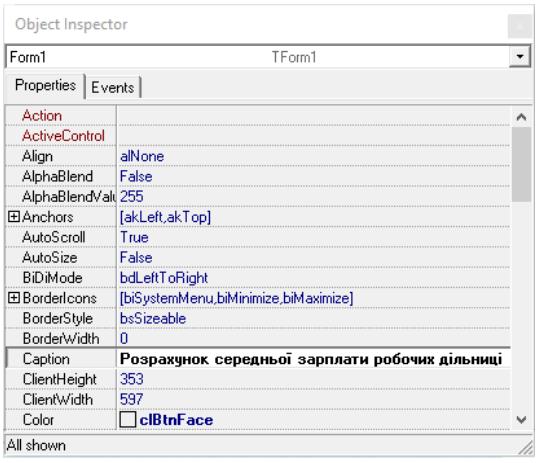


Рис. 2.9. Зміна значення властивості **Caption** (Заголовок) вікна форми «***Form1***» (Форма1) у вікні «***Object Inspector***» (Інспектор об'єктів) на **Розрахунок середньої зарплати робочих дільниці**

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.9) вибрати ім'я об'єкта (форми або компонентів) (табл. 2.1);

б) у лівій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де перелічені властивості об'єкта (форми чи компонентів) виділити властивість (табл. 2.1);

в) у правій колонці вкладки «***Properties***» (Властивості) (рис. 2.9), де вказано значення об'єкта (форми або компонентів) вибрати або ввести значення властивості (табл. 2.1).

2.4.2. Створення обробників подій

Створимо обробники подій для компонентів, вказаних у таблиці 2.2.

Таблиця 2.2 – Обробники подій компонентів для програми

Компонент	Подія	Ім'я обробника
btnCalculate	OnClick	btnCalculateClick
btnClear	OnClick	btnClearClick
btnExit	OnClick	btnExitClick
frmRezult	OnCreate	frmRezultCreate
GridInputfam	OnKeyPress	GridInputfamKeyPress
GridInputz1	OnKeyPress	GridInputz1KeyPress
GridInputz2	OnKeyPress	GridInputz2KeyPress
GridInputz3	OnKeyPress	GridInputz3KeyPress

Для цього необхідно виконати наступні дії:

а) у верхній частині вікна «***Object Inspector***» (Інспектор об'єктів) (рис. 2.10) вибрати ім'я об'єкта (форми або компонента) (табл. 2.2);

б) у цьому ж вікні потрібно вибрати вкладку «***Events***» (Події) (рис. 2.10);

в) у лівій колонці вкладки «***Events***» (Події) (рис. 2.10), де перелічені імена подій, які може використовувати вибраний об’єкт (компонент), вибрати ім'я події (табл. 2.2);

г) для створення обробника події потрібно у правій колонці вкладки «***Events***» (Події) навпроти обраної раніше події зробити подвійне клацання мишею у полі імені обробника (рис. 2.10);

д) відкриється вікно «***Unit_16.pas***» (Редактор коду), до якого буде додано

шаблон процедури обробника **btnExitClick** події **OnClick** (рис. 2.11), а у вікні «***Object Inspector***» (Інспектор об’єктів) поряд з ім'ям події з'явиться ім'я обробника (рис. 2.10);

е) у вікні «***Unit_16.pas***» (Редактор коду) між словами **begin** і **end** необхідно ввести інструкції, що реалізують обробку події (рис. 2.11). У лістингу програми, див. п. п 2.4.3, наведені тексти інструкцій: обробки події **OnClick** для кнопок: **Розрахувати** (btnCalculate), **Очистити** (btnClear) та **Вихід** (btnExit), а також обробки події **OnCreate** для вікна форми (frmRezult) та обробки події **OnKeyPress** для полів вводу компонентів (GridInputfam, GridInputz1, GridInputz2, GridInputz3).

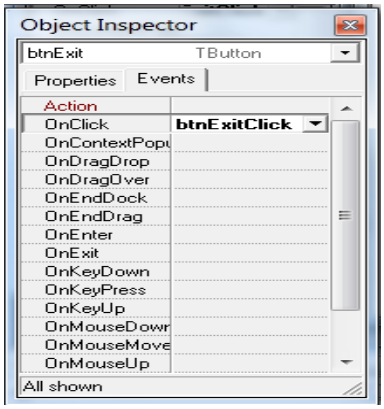


Рис. 2.10. Створення обробника **btnExitClick** події **OnClick** для компонента btnExit, на вкладці «***Events***» (Події) у вікні «***Object Inspector***» (Інспектор об'єктів)

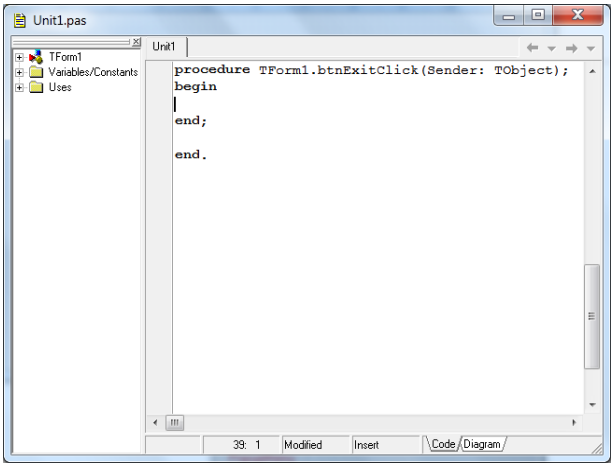


Рис. 2.11. Шаблон процедури обробника **btnExitClick** події **OnClick**, згенерований програмою Delphi

2.4.3. Створення вихідного коду

Вихідний код програми представлений у вигляді лістингу програми.

Лістинг програми

```
unit Unit_16;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
```

```
  Dialogs, StdCtrls, ExtCtrls, Grids;
type
  TfrmRezult = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    GridInputfam: TStringGrid;
    GridInputz1: TStringGrid;
    GridInputz2: TStringGrid;
    GridInputz3: TStringGrid;
    GridOutputSredz: TStringGrid;
    MemOutput: TMemo;
    BtnCalculate: TButton;
    BtnClear: TButton;
    BtnExit: TButton;
    procedure BtnExitClick(Sender: TObject);
    procedure BtnClearClick(Sender: TObject);
    procedure BtnCalculateClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure GridInputfamKeyPress(Sender: TObject; var Key: Char);
    procedure GridInputz1KeyPress(Sender: TObject; var Key: Char);
    procedure GridInputz2KeyPress(Sender: TObject; var Key: Char);
    procedure GridInputz3KeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
end;
   // Тип даних - запис
type Raboch = record
          fam : string;
           z1  : real;
           z2  : real;
           z3  : real;
        sredz : real;
```

```
end;
const
  n=5; // Кількість робітників
var
  frmRezult: TfrmRezult;
  // Оголошення масиву записів
  Work : array [1..n] of Raboch;
  i:integer;
implementation
{$R *.dfm}
procedure TfrmRezult.BtnExitClick(Sender: TObject);
begin
  Close;  // Закриття програми
end;
procedure TfrmRezult.BtnClearClick(Sender: TObject);
var
  i:integer;  // Оголошення змінних
begin
   // Очищення полів введення даних
  for i:=1 to n do
  begin
    GridInputfam.Cells[i,0]:=' ';
    GridInputz1.Cells[i,0]:=' ';
    GridInputz2.Cells[i,0]:=' ';
    GridInputz3.Cells[i,0]:=' ';
    // Очищення полів виведення середньої зарплати робітника
    GridOutputsredz.Cells[i,0]:=' ';
  end;
   // Очищення полів виводу
   memoutput.lines.Clear;
end;
procedure TfrmRezult.BtnCalculateClick(Sender: TObject);
var
  // Оголошення змінних
  i:integer;
```

```
begin
    // Очищення рядка
    memOutput.Lines.Clear;
    // Найменування колонок інформації, що виводиться
    memOutput.Lines.Add('Прізвище'+'       '+'Серед. зарп.');
   // Ввід даних до масиву
   for i:=1 to n do
   begin
    // Захист полів вводу
    if (GridInputfam.Cells[i,0]<>' ') and (GridInputz1.Cells[i,0]<>' ') and
    (GridInputz2.Cells[i,0]<>' ') and (GridInputz3.Cells[i,0]<>' ') then
    begin
     with Work[i] do
     begin
      fam:=GridInputfam.Cells[i,0];
      z1:=StrToFloat(GridInputz1.Cells[i,0]);
      z2:=StrToFloat(GridInputz2.Cells[i,0]);
      z3:=StrToFloat(GridInputz3.Cells[i,0]);
      // Визначення середньої зарплати робітника
      sredz:=(work[i].z1+work[i].z2+work[i].z3)/3;
      // Вивід середньої зарплати робітника
      GridOutputsredz.Cells[i,0]:= FloatToStrF(sredz,ffGeneral,5,2);
      // Вивід середньої зарплати всіх робітників
      memoutput.lines.Add(Work[i].fam+'     '+
      FloattoStrF((Work[i].sredz),ffGeneral,5,2));
     end;
    end
    else
     begin
      Application.MessageBox('У полях вводу даних порожньо.'#13#10
     'Введіть дані!', 'Помилка', mb_IconError + mb_OK);
      exit;
     end;
   end;
end;
```

```
procedure TfrmRezult.FormCreate(Sender: TObject);
begin
   // Найменування полів вводу
  GridInputfam.Cells[0,0]:='Прізвище';
  GridInputz1.Cells[0,0]:='Зарп.1 місяц';
  GridInputz2.Cells[0,0]:='Зарп.2 місяц';
  GridInputz3.Cells[0,0]:='Зарп.3 місяц';
  // Найменування полів виводу
  GridOutputsredz.Cells[0,0]:='Серед. зарп.';
end;
procedure TfrmRezult.GridInputz1KeyPress(Sender: TObject; var Key: Char);
begin
   case key of
    '0'..'9':;  // Дозволено ввід цифр від 0 до 9
    #8:; // Дозволено використання клавіші <Backspace>
    // Дозволено використання коми
    ',': begin
          // Заборона ввід другої коми
          if pos(',',GridInputz1.Cells[GridInputz1.Col,0])<>0 then key:=#0;
         end;
    else
     key:=#0;  // Ввід інших цифр та символів заборонено
    end;
end;
procedure TfrmRezult.GridInputz2KeyPress(Sender: TObject; var Key: Char);
begin
   case key of
    '0'..'9':;  // Дозволено ввід цифр від 0 до 9
    #8:;  // Дозволено використання клавіші <Backspace>
    // Дозволено використання коми
    ',': begin
         // Заборона ввід другої коми
         if pos(',',GridInputz2.Cells[GridInputz2.Col,0])<>0 then key:=#0;
         end;
    else
```

```
    key:=#0; // Ввід інших цифр та символів заборонено
   end;
end;
procedure TfrmRezult.GridInputz3KeyPress(Sender: TObject; var Key: Char);
begin
  case key of
   '0'..'9':;  // Дозволено ввід цифр від 0 до 9
   #8:;  // Дозволено використання клавіші <Backspace>
   // Дозволено використання коми
   ',': begin
        // Заборона ввід другої коми
        if pos(',',GridInputz3.Cells[GridInputz3.Col,0])<>0 then key:=#0;
        end;
   else
    key:=#0;  // Ввід інших цифр та символів заборонено
   end;
end;
procedure TfrmRezult.GridInputfamKeyPress(Sender: TObject; var Key: Char);
begin
 case key of
     'А'..'Я', 'а'..'я', 'І','Ї', 'і','ї','Є','є':;  // Дозволено ввід букв
     #8:;  // Дозволено використання клавіші <Backspace>
 else
   key:=#0;  // Ввід інших цифр та символів заборонено
 end;
end;
end.
```

2.4.4. Компіляція програми

Процес компіляції складається із двох етапів. На першому етапі виконується перевірка тексту програми на відсутність синтаксичних помилок, а на другому – генерується виконувана програма (exe-файл).

Компіляція програми виконується з «***Головного***» меню **Project** (Проєкт) обравши команду **Compile Project** (Компілювати проєкт) (рис. 2.12) або натиснути клавіші **Ctrl+F9**.

Повідомлення про помилки, попередження та підказки відображаються в нижній частині вікна редактора коду (рис. 2.13).

Процес усунення помилок має ітераційний характер. Зазвичай, спочатку усуваються найбільш очевидні помилки, наприклад, декларуються неоголошені змінні. Після чергового внесення змін до тексту програми виконується повторна компіляція. Слід зважати на те, що компілятор не завжди може точно локалізувати помилку. Тому аналізуючи фрагмент програми який на думку компілятора містить помилку, потрібно звертати увагу не тільки на той фрагмент коду, на який компілятор встановив курсор, але й на той, який знаходиться у попередньому рядку.

Якщо у програмі немає синтаксичних помилок, компілятор створює виконуваний файл програми. Ім'я виконуваного файлу таке ж, як і у файлу проєкту, а розширення – exe. Delphi поміщає виконуваний файл у той самий каталог, де знаходиться файл проєкту.

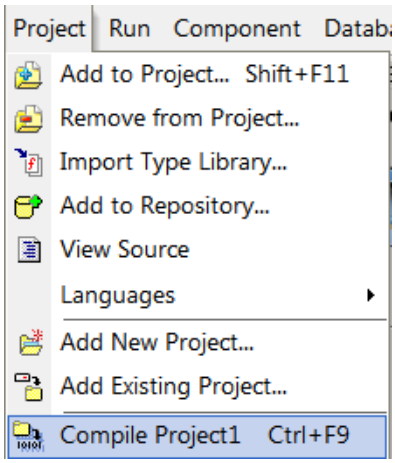


Рис. 2.12. Вибір команди **Compile Project** (Компілювати проєкт) з «***Головного***» меню **Project** (Проєкт)

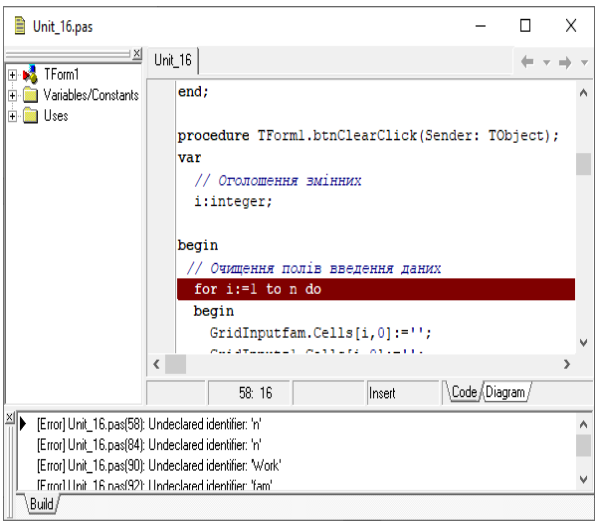


Рис. 2.13. Повідомлення компілятора про виявлені помилки

2.4.5. Запуск програми

Запуск програми може виконуватися з «***Головного***» меню **Run** (Запуск) обравши команду **Run** (Запуск) або натиснути на кнопці  **Run** (Запуск) на панелі інструментів «***Debug***» (Налагодження) (рис. 2.14) або натиснути клавішу **F9.**

Рис. 2.14. Панель інструментів «***Debug***» (Налагодження)

2.4.6. Помилки часу виконання

Під час роботи програми можуть виникати помилки, які називаються **помилками часу виконання** (run-time errors) або **винятками** (exceptions). У більшості випадків причинами винятків є невірні вихідні дані.

Після виникнення таких помилок програма перестане робити.

Необхідно перервати неробочу програму, вибравши із «***Головного***» меню **Run** (Запуск) команду **Program Reset** (Скидання програми) (рис. 2.15).

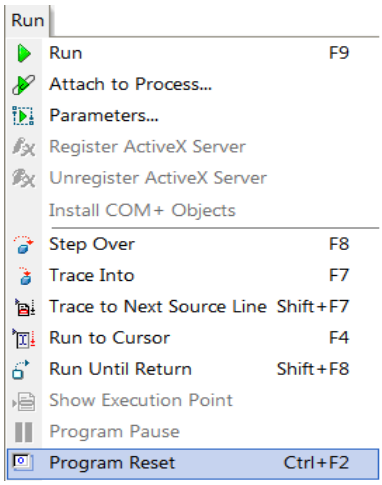


Рис. 2.15. Вибір команди **Program Reset** (Скидання програми) із «***Головного***» меню **Run** (Запуск)

2.4.7. Захист полів вводу

Для виключення помилок, при вводі даних у поля вводу необхідно передбачити у вихідному коді програми захист полів вводу.

У лабораторній роботі №16 передбачимо такі види захисту полів вводу (див. п. п. 2.4.3):

- захист полів вводу від вводу порожнього рядка;

- захист полів вводу від вводу літер та деяких символів, використовуючи подію OnKeyPress.

3. Зміст звіту з лабораторної роботи №16

Звіт з лабораторної роботи №16 складається з:

- титульного аркуша (див. Додаток Р);

- аркуша (див. Додаток Р) на якому розташовано: **тема** лабораторної роботи, **мета** лабораторної роботи, **розробка програми** (наведіть **умову**, яка розглядається в лабораторній роботі №16, див. п. п. 2.4);

- аркуша (див. Додаток Р) на якому розташовано: **створення інтерфейсу програми** (навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №16, див. п. п. 2.4.1);

- аркушів (див. Додаток Р) на яких розташовані: **лістинг програми** (навести **лістинг** програми, який розглядається в лабораторній роботі №16, див. п. п. 2.4.3) та **висновки**.

Запитання з самоконтролю до розділу 5

1. Що таке масив?
2. Що таке статичний масив?
3. Що таке динамічний масив?
4. Що таке одномірний масив?
5. Що таке багатомірний масив?
6. Що таке запис?
7. З чого складаэться запис?

Додаток І

Форма звіту з лабораторної роботи №9

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ

з лабораторної роботи №9
«Циклічні структури. Оператор for»
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав ст. гр. _____
(№ групи)
_____
(П.І.Б.)

Перевірив _____
(П.І.Б.)

Дніпро 20 ____

Тема:

Мета роботи:

Розробка програми
(навести **умову** та **формулу**, які розглядаються в лабораторній роботі №9, див. п. п. 2.4)

Створення інтерфейсу програми та виведення повідомлень програми
(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №9, див. п. п. 2.4.1)

Лістинг програми
(навести **лістинг** програми, який розглядається в лабораторній роботі №9, див. п. п. 2.4.3)

Висновки:

Додаток К

Форма звіту з лабораторної роботи №10

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ

з лабораторної роботи №10
«Циклічні структури. Оператор while»
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав ст. гр. _____
(№ групи)
_____
(П.І.Б.)

Перевірив _____
(П.І.Б.)

Дніпро 20 ____

Тема:

Мета роботи:

Розробка програми
(навести **умову** та **формулу**, які розглядаються в лабораторній роботі №10, див. п. п. 2.4)

Створення інтерфейсу програми та виведення повідомлень програми
(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №10, див. п. п. 2.4.1)

Лістинг програми
(навести **лістинг** програми, який розглядається в лабораторній роботі №10, див. п. п. 2.4.3)

Висновки:

Додаток Л

Форма звіту з лабораторної роботи №11

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ІНСТИТУТ ПРОМИСЛОВИХ ТА БІЗНЕС ТЕХНОЛОГІЙ
УКРАЇНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ
з лабораторної роботи №11
«Циклічні структури. Оператор repeat»
з дисципліни «Об'єктно-орієнтоване програмування

Виконав ст. гр. _____
(№ групи)
_____
(П.І.Б.)

Перевірив _____
(П.І.Б.)

Дніпро 20 ____

Тема:

Мета роботи:

Розробка програми

(навести **умову** та **формулу**, які розглядаються в лабораторній роботі №11, див. п. п. 2.4)

Створення інтерфейсу програми та виведення повідомлень програми

(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №11, див. п. п. 2.4.1)

Лістинг програми

(навести **лістинг** програми, який розглядається в лабораторній роботі №11, див. п. п. 2.4.3)

Висновки:

Додаток М

Форма звіту з лабораторної роботи №12

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ІНСТИТУТ ПРОМИСЛОВИХ ТА БІЗНЕС ТЕХНОЛОГІЙ
УКРАЇНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ
з лабораторної роботи №12
«Циклічні структури. Вкладені оператори»
з дисципліни «Об'єктно-орієнтоване програмування

Виконав ст. гр. _____
(№ групи)
_____
(П.І.Б.)
Перевірив _____
(П.І.Б.)

Дніпро 20 ____

Тема:

Мета роботи:

Розробка програми

(навести **умову** та **формулу**, які розглядаються в лабораторній роботі №12, див. п. п. 2.4)

Створення інтерфейсу програми та виведення повідомлень програми

(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №12, див. п. п. 2.4.1)

Лістинг програми

(навести **лістинг** програми, який розглядається в лабораторній роботі №12, див. п. п. 2.4.3)

Висновки:

Додаток Н

Форма звіту з лабораторної роботи №13

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ІНСТИТУТ ПРОМИСЛОВИХ ТА БІЗНЕС ТЕХНОЛОГІЙ
УКРАЇНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ
з лабораторної роботи №13
«Структури даних. Одномірний статичний масив»
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав ст. гр. _____
(№ групи)
_____
(П.І.Б.)

Перевірив _____
(П.І.Б.)

Дніпро 20 ____

Тема:

Мета роботи:

Розробка програми
(навести **умову**, яка розглядається в лабораторній роботі №13, див. п.п. 2.4)

Створення інтерфейсу програми та виведення повідомлень програми
(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №13, див. п.п. 2.4.1)

Лістинг програми
(навести **лістинг** програми, який розглядається в лабораторній роботі №13, див. п.п. 2.4.3)

Висновки:

Додаток О

Форма звіту з лабораторної роботи №14

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ІНСТИТУТ ПРОМИСЛОВИХ ТА БІЗНЕС ТЕХНОЛОГІЙ
УКРАЇНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ
з лабораторної роботи №14
«Структури даних. Двомірний статичний масив»
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав ст. гр. _____
(№ групи)
_____
(П.І.Б.)

Перевірив _____
(П.І.Б.)

Дніпро 20 ____

Тема:

Мета роботи:

Розробка програми
(навести **умову**, яка розглядається в лабораторній роботі №14, див. п.п. 2.4)

Створення інтерфейсу програми та виведення повідомлень програми
(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №14, див. п.п. 2.4.1)

Лістинг програми
(навести **лістинг** програми, який розглядається в лабораторній роботі №14, див. п.п. 2.4.3)

Висновки:

Додаток П

Форма звіту з лабораторної роботи №15

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ІНСТИТУТ ПРОМИСЛОВИХ ТА БІЗНЕС ТЕХНОЛОГІЙ
УКРАЇНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ
з лабораторної роботи №15
«Структури даних. Одномірний динамічний масив»
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав ст. гр. _____
(№ групи)
_____
(П.І.Б.)

Перевірив _____
(П.І.Б.)

Дніпро 20 ____

Тема:

Мета роботи:

Розробка програми
(навести **умову**, яка розглядається в лабораторній роботі №15, див. п.п. 2.4)

Створення інтерфейсу програми та виведення повідомлень програми
(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №15, див. п.п. 2.4.1)

Лістинг програми
(навести **лістинг** програми, який розглядається в лабораторній роботі №15, див. п.п. 2.4.3)

Висновки:

Додаток Р

Форма звіту з лабораторної роботи №16

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ІНСТИТУТ ПРОМИСЛОВИХ ТА БІЗНЕС ТЕХНОЛОГІЙ
УКРАЇНСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ НАУКИ І ТЕХНОЛОГІЙ

Кафедра «Технологія машинобудування»

ЗВІТ
з лабораторної роботи №16
«Структури даних. Тип даних – запис»
з дисципліни «Об'єктно-орієнтоване програмування»

Виконав ст. гр. _____
(№ групи)
_____
(П.І.Б.)

Перевірив _____
(П.І.Б.)

Дніпро 20 ____

Тема:

Мета роботи:

Розробка програми
(навести **умову**, яка розглядається в лабораторній роботі №16, див. п.п. 2.4)

Створення інтерфейсу програми та виведення повідомлень програми
(навести **інтерфейс** програми та **вивід повідомлень** програми, які розглядаються в лабораторній роботі №16, див. п.п. 2.4.1)

Лістинг програми
(навести **лістинг** програми, який розглядається в лабораторній роботі №16, див. п.п. 2.4.3)

Висновки:

ЛІТЕРАТУРА

Основна

1. Керман Митчелл К. Программирование и отладка в Delphi : учеб. курс. Москва : Вильямс, 2004. 720 с.

2. Дудзяний І. М. Програмування мовою Object Pascal : навч. посіб. Львів : Видавничий центр ЛНУ імені Івана Франка, 2003. 328 с.

3. Безменов М. І. Основи програмування у середовищі Delphi : навч. посіб. Харків : НТУ "ХПІ", 2010. 608 с.

4. Основи програмування. Delphi 6 : навч. посіб. / Алексеєв М. О., Кандзюба С. П., Коротенко Л. М., Шевцова О. С. Дніпропетровськ : Національний гірничий університет, 2013. 272 с.

5. Основи об’єктно-орієнтованого програмування : навч. посіб. / Семйон І. В., Чупов С. В., Брила А. Ю., Апшай Н. І. Ужгород, 2011. 141 с.

Додаткова

6. Алхімова С. М. Об’єктно-орієнтоване програмування : підручник : у 2-х ч. Київ : КПІ ім. Ігоря Сікорського, 2019. Ч. 2 : Об’єктно-орієнтований підхід до розробки програмного забезпечення. 192 с.

7. Омельчук Л. Л. Об’єктно-орієнтоване програмування. Лабораторний практикум : навч. посіб. Київ, 2021. 265 с.

8. Лабораторний практикум з програмування : навч. посіб. / за заг. ред. проф. А. П. Власюка. Рівне : НУВГП, 2011. 495 с.

Інформаційні ресурси в Інтернеті

9. Для чайників. Pascal. Вступ. Алгоритм. Структура програми. Компілятор., 2020. *YouTube*. URL: https://www.youtube.com/watch?v=bws2rE6bNew (дата звернення: 10.02.2024).

10. Embarcadero Cross-Platform App Development Software. *Embarcadero*. URL: https://www.embarcadero.com/ (date of access: 10.02.2024).

Навчально-методичне видання

Гришин Володимир Сергійович,
Карабут Владлен Миколайович

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Навчально-методичні рекомендації
до виконання лабораторного практикуму
Частина ІІ

Електронне видання

Експертний висновок склав д-р техн. наук, проф. Володимир Анісімов

Зареєстровано НМВ УДУНТ (№ 707 від 13.03.2024)

В авторській редакції
Комп’ютерна верстка Карабут В. М.

Формат 60×84 1/16. Ум. друк. арк. 7,56. Обл.-вид. арк. 3,27.
Зам. № 31

Видавець: Український державний університет науки і технологій
вул. Лазаряна, 2, ауд. 2216, м. Дніпро, 49010.
Свідоцтво суб’єкта видавничої справи ДК № 7709 від 14.12.2022

Адреса видавця та дільниці оперативної поліграфії:
вул. Лазаряна, 2, Дніпро, 49010