

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Івано-Франківський національний технічний
університет нафти і газу**

Кафедра комп'ютеризованого машинобудівного виробництва

В. Б. Копей

ОСНОВИ ПРОГРАМУВАННЯ

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ**

**Івано-Франківськ
2018**

УДК 004.43
К 65

Рецензент:

Онисько О. Р., кандидат технічних наук, доцент,
доцент кафедри комп'ютеризованого
машинобудівного виробництва Івано-Франківського
національного технічного університету нафти і газу

*Рекомендовано методичною радою університету
(протокол № 7 від 29 травня 2018 р.)*

К65 Копей В. Б. Методичні вказівки до виконання курсової
роботи з дисципліни "Основи програмування" / В. Б. Копей -
Івано-Франківськ: ІФНТУНГ, 2018. - 28 с.

МВ 02070855-11481 - 2018

Методичні вказівки призначені для виконання курсової роботи з дисципліни "Основи програмування". Описано зміст курсової роботи, подано рекомендовані теми курсових робіт та код програм з прикладами використання основних елементів мови Delphi і основних компонентів VCL. Розроблено відповідно до робочої програми дисципліни "Основи програмування" для підготовки бакалаврів за спеціальністю 131 - Прикладна механіка.

УДК 004.43

МВ 02070855-11481 - 2018

© Копей В. Б., 2018
© ІФНТУНГ, 2018

ЗМІСТ

Вступ	4
1 Зміст курсової роботи	5
2 Рекомендовані теми курсових робіт	11
3 Перелік рекомендованих джерел	12
Додаток А. Приклад з основними елементами мови Delphi ..	14
Додаток Б. Приклад з основними компонентами VCL	23

ВСТУП

Delphi (раніше Object Pascal) - імперативна структурована об'єктно-орієнтована мова програмування зі статичною типізацією змінних, яка використовується для створення різноманітного прикладного програмного забезпечення. Delphi основана на мові Паскаль (англ. Pascal) - одній з найбільш відомих мов програмування. Паскаль часто використовується для навчання програмуванню завдяки чіткій алгоритмічній структурі програми. Студент, який освоїв Паскаль, може з легкістю вивчити і інші алгоритмічні мови програмування.

Embarcadero Delphi (раніше Borland Delphi) - інтегроване середовище розробки програмного забезпечення мовою Delphi, створене спочатку фірмою Borland і на даний момент належить і розробляється Embarcadero Technologies. Для виконання курсової роботи рекомендується використовувати Borland Delphi 7, яке є найбільш популярним продуктом Borland, в основному, завдяки стабільності, швидкості і низьких вимог до апаратного забезпечення.

Відкритою альтернативою Embarcadero Delphi є Lazarus [1] - середовище розробки програмного забезпечення мовою Object Pascal для компілятора Free Pascal. Lazarus підтримує перетворення проектів Delphi, але немає з ним повної сумісності.

Метою курсової роботи є освоєння принципів сучасного об'єктно-орієнтованого і візуального програмування шляхом створення прикладного програмного забезпечення мовою Delphi.

В методичних вказівках описано зміст курсової роботи, подано рекомендовані теми курсових робіт, список рекомендованої літератури для вивчення основ мови Delphi [2-7] і її глибшого освоєння [8-24]. В додатках наведено код програм з прикладами використання основних елементів мови Delphi (додаток А) і основних компонентів VCL (додаток Б). Цей код може бути використаний як міні-довідник програміста мовою Delphi.

1 ЗМІСТ КУРСОВОЇ РОБОТИ

Стадії розробки програмного забезпечення регламентуються Єдиною Системою Програмної Документації (ЄСПД) ГОСТ 19.102-77. В курсовій роботі рекомендується дотримуватись наступного змісту: технічне завдання, опис алгоритму, опис програми, тестування і демонстрація програми, висновки, перелік використаних джерел, додаток з кодом програми.

Технічне завдання може містити:

- 1 Призначення і область використання програми.
- 2 Обґрунтування необхідності розробки програми.
- 3 Джерела, на основі яких ведеться розробка.
- 4 Вимоги до програми і програмної документації (вимоги до функціональних характеристик, вхідних і вихідних даних, надійності, інформаційній і програмній сумісності, вимоги до технічних засобів).

5 Техніко-економічні показники програми.

Опис алгоритму. В цьому розділі описується загальна структура програми, її принципи функціонування і її алгоритм. Алгоритм - це послідовність інструкцій, що описують порядок дій виконавця для досягнення певного результату. Алгоритм програми або її частин може бути описаний словесно (псевдокодом) або графічно (ГОСТ 19.701-90). Для прикладу, нижче наведено словесний і графічний (рис. 1) алгоритми процедури testIf (додаток А):

```
початок
ввести x
якщо 0<x<=10
    вивести x*2
інакше
    якщо 10<x<=100
        вивести x*3
    інакше
        вивести 0
кінець
```

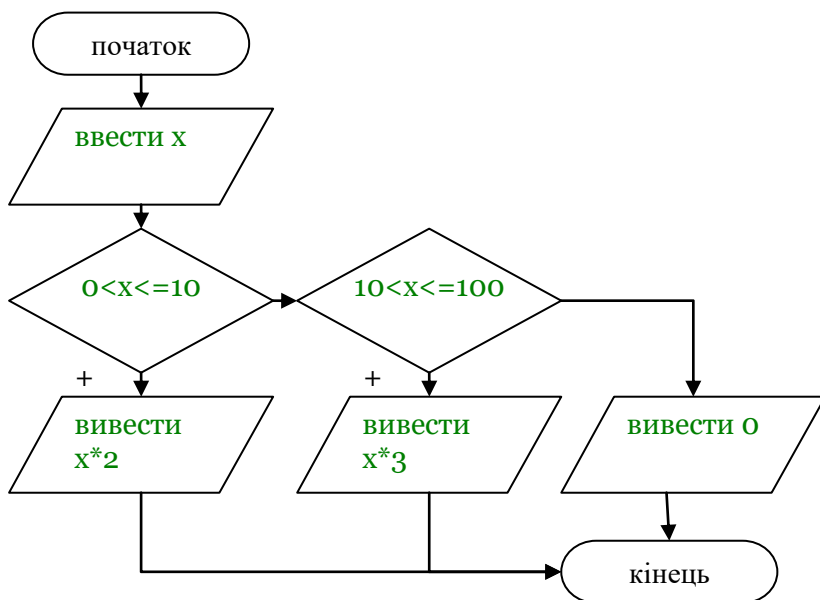


Рисунок 1 - Графічний алгоритм процедури testIf (додаток А)

Нижче наведено словесний і графічний (рис. 2) алгоритми процедури testFor (додаток А):

```

початок
для значень i від 0 до 5 з кроком 1
    вивести i
кінець
  
```

Опис програми. Тут вказується програмне забезпечення, яке необхідне для функціонування програми, мови програмування, на яких написана програма, мінімальні вимоги до технічних засобів. Сирцевий код усіх модулів програми наводиться в додатках. Код повинен містити детальні коментарі для усіх важливих команд. Сучасні підходи до створення якісного програмного коду викладені в праці [25]. Опис логічної структури програми повинен містити

опис модулів (табл. 1), класів (табл. 2), подій об'єктів (табл. 3), змінних, полів, функцій і методів (табл. 4), бібліотечних підпрограм (табл. 5), вхідних і вихідних даних, графічного інтерфейсу користувача (рис. 3).

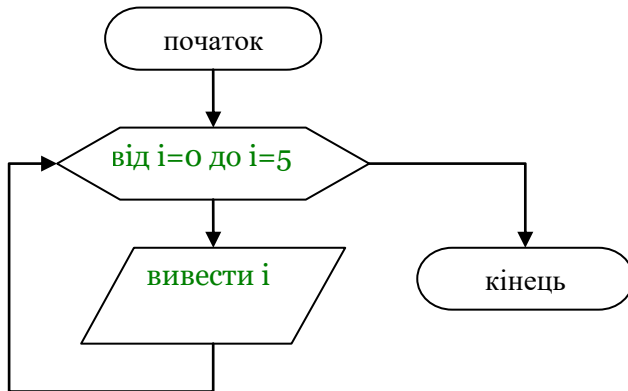


Рисунок 2 - Графічний алгоритм процедури testFor (додаток А)

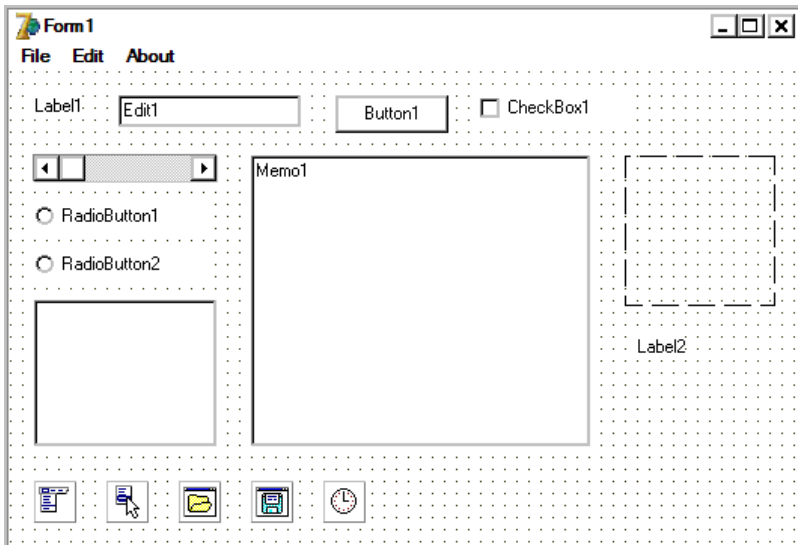


Рисунок 3 - Вигляд вікна на етапі проектування

Таблиці 1-5 заповнюються шляхом аналізу вихідного коду програми. Обробники подій об'єктів (табл. 3) наведено у вкладці Events вікна Object Inspector. В таблиці 4 описуються усі глобальні і локальні (для кожної підпрограми) типи, змінні, об'єкти, функції, константи, які визначені користувачем. Для заповнення таблиці 5 можна користуватись підказкою коду і пошуком документації Topic Search, яка в середовищі Delphi 7 викликається шляхом вибору в коді потрібного ідентифікатора і натиску клавіші F1 (рис. 4).

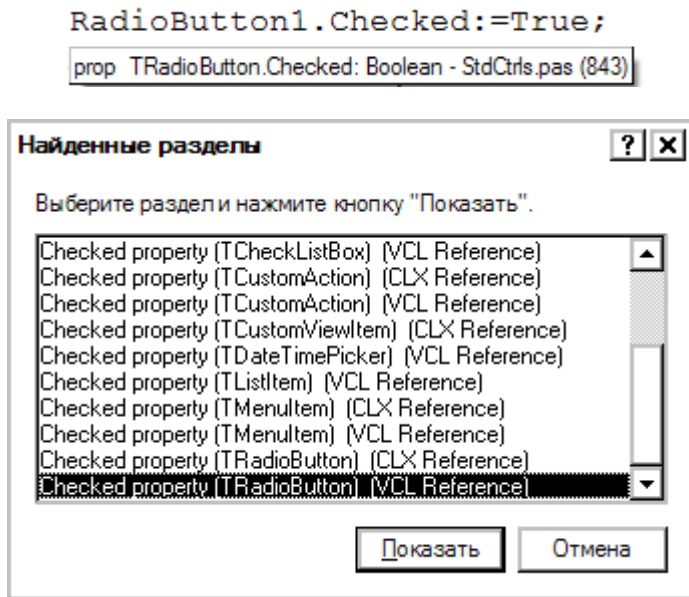


Рисунок 4 - Підказка коду і пошук документації

Таблиця 1 - Модулі програми

Модуль	Призначення
Project1.dpr	Головний модуль
Unit1.pas	Модуль з описом класу TForm1
...	...

Таблиця 2 – Опис членів класу TForm1

Тип	Назва, параметри	Призначення
	Поля:	
TButton	Button1	Кнопка на вікні
...
	Методи:	
-	Button1Click(Sender: TObject)	Викликається під час натиску на кнопку Button1
...

Таблиця 3 – Обробники подій об'єктів

Об'єкт	Подія	Обробник події
Button1	OnClick	Button1Click
...

Таблиця 4 - Визначені користувачем типи, змінні, об'єкти, функції, константи (глобальні і локальні для кожної підпрограми)

Тип	Назва, параметри	Призначення
	Глобальні	
клас	TForm1	Клас успадковує TForm і описує вікно програми
TForm1	Form1	Об'єкт-вікно
...
	Локальні для процедури TForm1.Button1Click	
real	x	Змінна для збереження дійсного числа
...

Таблиця 5 - Стандартні бібліотечні процедури, функції, константи, методи і властивості, які використовувались в програмі

Тип	Назва, параметри	Призначення
Boolean	property TRadioButton.Checked	Визначає чи вибраний перемикач
Integer	function TApplication.MessageBox(const Text, Caption: PChar; Flags: Longint = MB_OK	Відображає вказане повідомлення користувачу
...

Програма повинна містити коротке керівництво користувача (як правило доступне через меню Help), в якому описано призначення програми, умови виконання програми, її графічний інтерфейс (рис. 5), принципи роботи з програмою.

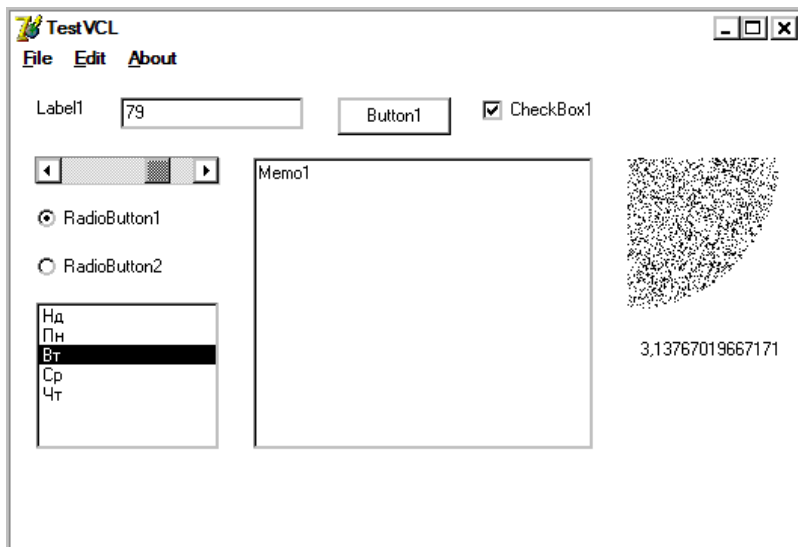


Рисунок 5 - Графічний інтерфейс програми

Тестування програми. Тут описується порядок і методика тестування програми. Часто помилки виконання виявляються за допомогою вбудованого налагоджувача. Виконується покрокове виконання програми або виконання програми до заданої точки з переглядом і модифікацією значень змінних. Для перехоплення помилок виконання в програмі слід застосовувати блоки `try-except`. Виявлення логічних помилок можливе шляхом тестування програми і за допомогою налагоджувача. Під час виконання програми слід контролювати завантаження процесору та об'єм оперативної пам'яті.

Висновки. У висновках слід сформулювати основні особливості розробленого програмного забезпечення, його переваги і недоліки у порівнянні з аналогами, шляхи удосконалення програми.

Перелік використаних джерел. В перелік використаних джерел входять літературні, Internet та інші джерела, які використовувались в роботі. В тексті пояснювальної записки обов'язковим є посилання на використані джерела в квадратних дужках (наприклад [30]).

2 РЕКОМЕНДОВАНІ ТЕМИ КУРСОВИХ РОБІТ

- 1 Програма для перегляду тривимірних моделей.
- 2 Компоненти VCL "індикатори".
- 3 Програма для розрахунку режимів різання.
- 4 Програма для розрахунку припусків на механічну обробку.
- 5 Програма-довідник по ЄСДП.
- 6 Редактор коду для токарних верстатів з ЧПК.
- 7 Симулятор токарного верстата з ЧПК.
- 8 Програма для аналізу точності механічної обробки.
- 9 Класи для операцій з матрицями.
- 10 Програма для візуалізації клітинних автоматів.
- 11 Мовний інтерпретатор.
- 12 Бібліотека класів з функціями математичної статистики.
- 13 Довідник з періодичної системи хімічних елементів.
- 14 Система управління базою знань.

- 15 Програма для візуалізації роботи ГВС.
- 16 Редактор електронних таблиць.
- 17 Графічний редактор векторної графіки.
- 18 Програма для побудови графів.
- 19 Програма для розв'язування диференційних рівнянь.
- 20 Програма для інтегрування і диференціювання функцій.
- 21 Програма для розв'язування системи лінійних рівнянь.
- 22 Програма для розв'язування рівнянь чисельними методами.
- 23 Програма для побудови графіків функцій.
- 24 Програма "органайзер".
- 25 Програма "науковий програмований калькулятор".
- 26 Прикладна програма для Excel.
- 27 Прикладна програма для Kompas 3D.
- 28 Прикладна програма для SolidWorks.
- 29 Програма для конвертації одиниць вимірювання.
- 30 Програма для симуляції руху математичного маятника.
- 31 Програма для розрахунку розмірних ланцюгів.
- 32 Програма для автоматизації підготовки операційних карт.

3 ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

- 1 Lazarus Homepage. Режим доступу: <http://www.lazarus-ide.org>
- 2 Глинський Я. М. Паскаль. Turbo Pascal і Delphi. 3-є вид. / Я. М. Глинський, В. Є. Анохін, В. А. Ряжська - Львів: "Деол", 2002. -144 с.
- 3 Онисько О. Р. Основи програмування : навчальний посібник. — Івано—Франківськ: ІФНТУНГ, 2017. — 156 с.
- 4 Культин Н . Б . Основы программирования в Delphi 7. — СПб .: БХВ - Петербург , 2007. — 608 с .: ил.
- 5 Парижский С. М. Delphi. Учимся на примерах / Под ред. Ю. А. Шпака - К.: "МК-Пресс", 2005.-216с, ил.
- 6 Справочник - "Основы Delphi". Режим доступа: <http://delphibasics.ru>
- 7 Осипов Д. Л. Delphi. Программирование для Windows, OS X, iOS и Android. — СПб.: БХВ-Петербург, 2014. — 464 с.: ил.
- 8 Архангельский А.Я. Delphi 2006. Справочное пособие: Язык Delphi, классы, функции Win32 и .NET. — М.: ООО «Бином-

- Пресс», 2006 г. — 1152 с.: ил.
- 9 Архангельский А.Я. Приемы программирования в Delphi на основе VCL. — М.: ООО «Бином-Пресс», 2006 г. — 944 с.: ил.
 - 10 Архангельский А. Я. Delphi 7. Справочное пособие. — М.: ООО «Бином-Пресс», 2004 г. — 1024 с.: ил.
 - 11 А.Я. Архангельский Программирование в Delphi 7. — М.: ООО «Бином-Пресс», 2003 г. — 1152 с.: ил.
 - 12 Фленов М.Е. Библия Delphi. - СПб.: БХВ-Петербург, 2004. - 880 с.: ил.
 - 13 Флёнов М. Е. Delphi 2005. Секреты программирования (+CD). — СПб.: Питер, 2006. — 266 с: ил.
 - 14 Фленов М. Е. Программирование в Delphi глазами хакера. — СПб.: БХВ-Петербург, 2003. - 368 с: ил.
 - 15 Корняков В. Н. Программирование документов и приложений MS Office в Delphi. — СПб.: БХВ-Петербург, 2005. - 496 с : ил.
 - 16 Сорокин А.В. Delphi. Разработка баз данных. - СПб.: Питер, 2005. - 477с.: ил.
 - 17 Кузан Д. Я. Программирование Win32 API в Delphi / Д. Я. Кузан, В. Н. Шапоров — СПб.: БХВ-Петербург, 2005. —368 с: ил.
 - 18 Ревич Ю. В. Нестандартные приемы программирования на Delphi. — СПб.: БХВ-Петербург, 2005. — 560 с: ил.
 - 19 Дарахвелидзе П. Г. Delphi 2005 для Win32 / П. Г. Дарахвелидзе, Е. П. Марков - СПб.: БХВ-Петербург, 2005. - 1136 с: ил.
 - 20 Марков Е. П. Delphi 2005 для .NET. / Е. П. Марков, В. В. Никифоров - СПб.: БХВ-Петербург, 2005. - 896 с : ил.
 - 21 Окулов С. М. Программирование в алгоритмах / С. М. Окулов. — М.: БИНОМ. Лаборатория знаний, 2002. — 341 с: ил.
 - 22 Зубов А. Программирование на Delphi. Трюки и эффекты (+CD). — СПб.: Питер, 2005. —396 с.: ил.
 - 23 Бакнелл Джулиан М. Фундаментальные алгоритмы и структуры данных в Delphi: Пер. с англ. / Джулиан М. Бакнелл. - СПб: ООО "ДиаСофтЮП", 2003. - 560 с.
 - 24 Фаронов В. В. Искусство создания компонентов Delphi (+CD). Библиотека программиста. - СПб.: Питер, 2005. - 463 с.: ил.
 - 25 Макконнелл С. Совершенный код. Мастер-класс / Пер. с англ. — М. : Издательство «Русская редакция», 2010. — 896 стр. : ил.

ДОДАТОК А. ПРИКЛАД З ОСНОВНИМИ ЕЛЕМЕНТАМИ МОВИ DELPHI

```
program Project1; // заголовок програми
// Для детальної довідки див. delphibasics.ru

{$APPTYPE CONSOLE} // директива компілятора
uses // приєднання модулів
    SysUtils, Math;

label // оголошення міток
    label1; // мітка (використання оператора goto не
// рекомендується)

type // опис типів
    color=(blue, green, yellow, orange, red); //
// перерахований тип
    month=1..12; // діапазонний тип
    Persona=record // тип-запис
        name:string[20]; // поле рядкового типу
        year:1900..2100; // поле діапазонного типу
        address:string[20]; // поле рядкового типу
    end;
    Line=class // тип-клас
        x1,y1:real; // поля класу
        x2,y2:real; // поля класу
        constructor Create(x1,y1,x2,y2: real); // конструктор
// класу
        function getLen():real; // метод класу
        procedure setLen(L:real); // метод класу
        property len:real read getLen write setLen; //
// властивість класу
    end;
```

```

const // оголошення констант
a1=5; // константа
s1='Hello world!'; // константа
a2:real=2.5; // типізована константа
ar1: array[0..4] of integer=(1,2,3,4,5); // константа-масив
масив
ar2: array[0..1,0..2] of integer=((1,2,3),(4,5,6)); //
константа-масив
set1=[blue, green]; // константа-множина

```

```

var // оголошення змінних
i:byte; // типу байт
k,n: integer; // цілого типу
x,y: real; // дійсного типу
c: char; // символьного типу
b: boolean; // булевого типу
s2: string; // рядкового типу
s3: string[10]; // рядкового типу
color1: color; // типу color (див. вище)
ar3: array[0..4] of integer; // масив цілих
ar4: array[0..1,0..2] of integer; // двовимірний масив
цілих
ar5: array of real; // відкритий масив дійсних
set2:set of color; // множина елементів типу color
set3:set of 1..100; // множина цілих від 1 до 100
file1: file of persona; // файл даних типу persona
file2: text; // текстовий файл
p: persona; // запис persona
line1: Line; // об'єкт класу Line

```

```

//-----
// реалізації методів класу Line:
constructor Line.Create(x1,y1,x2,y2: real); //
конструктор установлює значення полів

```

```

begin
    self.x1:=x1; self.y1:=y1; self.x2:=x2; self.y2:=y2;
end;

function Line.getLen():real; // метод-функція повертає
довжину лінії
begin
    getLen:=sqrt(sqr(x2-x1)+sqr(y2-y1)); {або Result:=}
end;

procedure Line.setLen(L:real); // метод-функція
установлює довжину лінії
var L0,dx0,dy0,dx,dy:real;
begin
    L0:=getLen(); {або L0:=self.getLen();} // попередня
довжина
    dx0:=x2-x1; dy0:=y2-y1; // попередні проекції лінії
    dx:=dx0*L/L0; dy:=dy0*L/L0; // нові проекції
    x2:=x1+dx; y2:=y1+dy; // нові координати точки 2
end;

//-----
procedure testTypes; // приклади присвоювання змінним
значень різного типу
begin
    i:=255; // ціле
    k:=32767; // ціле
    x:=13.5e-5; // дійсне
    c:='A'; // символ
    b:=true; // булеве (true або false)
    b:=i<>10; // булеве
    s2:='Hello'; // рядок
    color1:=green; // типу color
    ar3[0]:=5; // присвоїти першому елементу масиву ціле

```


значення

end;

procedure testIf; // приклад оператора умови if-then-else
begin

 readln(x); // ввести x

if (x>0) **and** (x<=10) **then** // якщо $0 < x \leq 10$ тоді

 writeln(x*2) // вивести $x \cdot 2$

else if (x>10) **and** (x<=100) **then** // якщо $10 < x \leq 100$
 тоді

 writeln(x*3) // вивести $x \cdot 3$

else // інакше

 writeln(0); // вивести 0

end;

procedure testCase; // приклад оператора умови case-of-else

begin

 k:=5; // присвоїти k значення

case k **of** // у випадку якщо

 1: y:=3; // k=1 то y:=3

 2: y:=6; // k=2 то y:=6

 3,4: y:=7; // k=3 або 4 то y:=7

 5..9: y:=10; // k=5..9 то y:=10

else y:=0; // у інших випадках y:=0

end;

end;

procedure testFor; // приклад оператора циклу for-to-do

var i:byte;

begin

for i:=0 **to** 5 **do** // повторювати від i=0 до i=5

 writeln(i) // цю команду

end;

```

procedure testWhile; // приклад оператора циклу while-do
begin
  i:=0;
  while i<=5 do // поки i<=5 повторювати
  begin // цей блок команд
    writeln(i);
    i:=i+1;
  end
end;

```

```

procedure testRepeat; // приклад оператора циклу repeat-
until
begin
  i:=0;
  repeat // повторювати наступні команди
    writeln(i);
    i:=i+1;
  until i>5; // поки i не більше 5
end;

```

```

// приклад оголошення процедури
// з параметрами-аргументами a,x і параметром-результатом
у

```

```

procedure testProcedure(a,x: real; var y: real);
var k:real; // локальна змінна
begin
  k:=a+1;
  y:=power(x,k);
end;

```

```

// приклад оголошення функції дійсного типу
// з параметром x
function testFunction(x:real):real;

```

```

const a1=2; // локальна константа
begin
    testFunction:=sqrt((a1*sin(x)-1)/abs(x+1)); //
    функція повертає це значення
    //або Result:=
end;

procedure testArray; // приклад роботи з масивом
var i:byte;
begin
    SetLength(ar5,high(ar1)); // установити довжину масиву
    for i:=0 to high(ar1) do // для кожного значення індексу
    масиву ar1
        begin // виконувати ці команди
            ar5[i]:=ar1[i]+1; // присвоїти значення i-му елементу
            масиву ar5
            writeln(ar5[i]); // вивести його
        end;
end;

procedure testArray2D; // приклад роботи з двовимірним
    масивом
var i,j:byte;
begin
    for i:=0 to high(ar2) do // для кожного рядка
        begin
            for j:=0 to high(ar2[0]) do // для кожного стовпчика
                begin
                    ar4[i,j]:=ar2[i,j]+1;
                    write(ar4[i,j], ' '); // вивести елемент в i-рядку та j-
                    стовпчику
                end;
                writeln; // вивести кінець рядка масиву
            end;
        end;

```

end;

procedure testString; // приклад роботи з рядками

var i:byte;

begin

for i:=1 **to** length(s1) **do** // для кожного індексу символу

 writeln(s1[i]); // вивести символ рядка з індексом i

 s2:=s1+'world'; // об'єднати рядки

 s2:=copy(s1,7,5); // копіювати 5 символів s1 починаючи з

7

 i:=pos('world', s1); // позиція підрядка в рядку s1

 insert('Hello',s2,0); // вставити в рядок s2 починаючи з 0

 delete(s2,2,1); // видалити 1 символ починаючи з 2

 str(2.5,s2); // конвертувати число 2.5 в рядок s2

 val(s2,x,k); // конвертувати рядок s2 в число x

end;

procedure testSet; // приклади роботи з множинами

begin

 set2:=[red,blue]; // множина

 set3:=[1,2,3]; // множина

 set3:=[1,2]+[3]; // об'єднання множин

 set3:=[1,2,3]*[3,4]; // перетин множин

 set3:=[1,2,3]-[2,3]; // різниця множин

 writeln([1,2,3]=[1,2]); // порівняння множин

 writeln(2 in [1,2,3]); // чи елемент 2 в множині?

end;

procedure testRecord; // приклади роботи з записами

begin

 p.name:='Ivanov'; // присвоїти значення полю name

 p.year:=1990;

 p.address:='Franka str, 15';

 writeln(p.name,'; ',p.year,'; ',p.address); // вивести

значення полів

end;

procedure testClass; // приклади роботи з об'єктами класу
begin

line1:=Line.Create(0,0,1,1); // створити об'єкт шляхом
виклику конструктора

line1.len:=10; // присвоїти значення властивості (див.
метод setLen)

writeln('x2=', line1.x2, ' y2=', line1.y2); // вивести
значення полів

end;

procedure testFile; // приклад роботи з файлом
begin

testRecord; // створити запис
assign(file1, 'myfile.dat'); // пов'язати файлову змінну з
файлом

rewrite(file1); // відкрити файл для запису

write(file1,p); // записати у файл запис p

close(file1); // закрити файл

reset(file1); // відкрити файл для читання

read(file1,p); // прочитати з файлу запис p

write(eof(file1)); // чи досягнуто кінця файлу?

end;

procedure testTextFile; // приклад роботи з текстовим
файлом

begin

assign(file2, 'myfile.txt'); // пов'язати файлову змінну з
файлом

rewrite(file2); // відкрити файл для запису

writeln(file2,'Hello world!'); // записати у файл рядок

close(file2); // закрити файл

```

append(file2); // відкрити файл для додання
writeln(file2,'Привіт світ!'); // записати у файл рядок
close(file2);
reset(file2); // відкрити файл для читання
readln(file2,s2); // прочитати з файлу рядок
write(s2);
end;

//-----
// виконувана частина
// розкоментуйте потрібний рядок, відкомпілюйте і
// виконайте програму
begin
testTypes;
//testIf;
//testCase;
//testFor;
//testWhile;
//testRepeat;
//testProcedure(2,2,y); writeln('y=',y);
//y:=testFunction(2); writeln('y=',y:2:3);
//testArray;
//testArray2D;
//testString;
//testSet;
//testRecord;
//testClass;
//testFile;
//testTextFile;
readln // очікує натиску клавіші Enter
end.

```

ДОДАТОК Б. ПРИКЛАД З ОСНОВНИМИ КОМПОНЕНТАМИ VCL

unit Unit1; // модуль Unit1

interface // інтерфейсна частина модуля

uses // використовувати модулі

Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
Dialogs, Menus, StdCtrls, ExtCtrls, Math;

type

// клас TForm1 успадковує TForm і описує вікно програми

TForm1 = **class**(TForm)

// члени класу (див. вікно Form1 та реалізації методів):

MainMenu1: TMainMenu;

PopupMenu1: TPopupMenu;

Label1: TLabel;

Edit1: TEdit;

Memo1: TMemo;

Button1: TButton;

CheckBox1: TCheckBox;

RadioButton1: TRadioButton;

RadioButton2: TRadioButton;

ListBox1: TListBox;

ScrollBar1: TScrollBar;

File1: TMenuItem;

Edit2: TMenuItem;

About1: TMenuItem;

Open1: TMenuItem;

Save1: TMenuItem;

Exit1: TMenuItem;

```

OpenDialog1: TOpenDialog;
SaveDialog1: TSaveDialog;
InsertDate1: TMenuItem;
InsertDate2: TMenuItem;
PaintBox1: TPaintBox;
Timer1: TTimer;
Label2: TLabel;
procedure Open1Click(Sender: TObject);
procedure Save1Click(Sender: TObject);
procedure About1Click(Sender: TObject);
procedure InsertDate1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure ListBox1DbClick(Sender: TObject);
procedure ScrollBar1Change(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
private // члени, які доступні тільки в класі
{ Private declarations }
public // загальнодоступні члени
a:real;
k,n:integer;
{ Public declarations }
end;

var
Form1: TForm1; // об'єкт-вікно класу TForm1

implementation // частина реалізації модуля

{$R *.dfm} // директива приєднує файл визначення форми

// викликається першою під час створення вікна

```



```

procedure TForm1.FormCreate(Sender: TObject);
const A: array[0..4] of string=('Нд','Пн','Вт','Ср','Чт');
// константа-масив
var i: integer;
begin
    // ініціалізація полів об'єкта класу TForm1
    self.a:=2; // тут self необхідне, оскільки є локальна A
    k:=1; // тут self не обов'язкове
    n:=1;
    Caption:='TestVCL';
    // ініціалізація властивостей інших об'єктів
    RadioButton1.Checked:=True;
    Memo1.PopupMenu:=PopupMenu1;
    InsertDate2.OnClick:=InsertDate1Click;
    for i:=0 to high(A) do // заповнити ListBox1
        ListBox1.Items.Add(A[i]);
end;

// вибір меню Open
procedure TForm1.Open1Click(Sender: TObject);
begin
    OpenFileDialog1.FileName := '*.txt';//показувати в діалозі
    тільки такі файли
    if OpenFileDialog1.Execute() then//показати діалог і, якщо
    вибір файлу зроблений
        //прочитати рядки з вибраного файлу
        Memo1.Lines.LoadFromFile(OpenFileDialog1.FileName);
end;

// вибір меню Save
procedure TForm1.Save1Click(Sender: TObject);
begin
    if SaveDialog1.Execute() then//показати діалог і, якщо
    вибір файлу зроблений

```

```

    //записати рядки у вибраний файл
    Memo1.Lines.SaveToFile(SaveDialog1.FileName);
end;

// вибір меню About
procedure TForm1.About1Click(Sender: TObject);
const s='TestVCL v0.1'+#13#10+'author: Kopey
V.'+#13#10+'vkohey@gmail.com';
begin
    Application.MessageBox(s,'Про програму');
end;

// вибір меню InsertDate
procedure TForm1.InsertDate1Click(Sender: TObject);
var pos:integer; s: string;
begin
    s:=Memo1.Lines.Text; // весь текст у Memo1
    pos:=Memo1.SelStart; // позиція курсору
    insert(DateToStr(Date),s,pos+1); // вставити дату в
    позицію курсору
    Memo1.Lines.Text:=s; // установити текст у Memo1
end;

// вибір меню Exit
procedure TForm1.Exit1Click(Sender: TObject);
begin
    Form1.Close; // закрити вікно
end;

// натиск кнопки Button1
procedure TForm1.Button1Click(Sender: TObject);
var x:real;
begin
    try // блок перехоплення можливих помилок

```

// щоб цей блок спрацював, виконайте програму поза Delphi IDE

x:=StrToFloat(Edit1.Text); // конвертувати рядок в число

Edit1.Text:=FloatToStr(x*a);

except // якщо помилка

ShowMessage('Введіть в Edit1 число!');

end;

end;

// подвійний натиск на список ListBox1

procedure TForm1.ListBox1DbClick(Sender: TObject);

var i:integer; s:string;

begin

i:=ListBox1.ItemIndex; //індекс вибраного елементу

s:=ListBox1.Items.Strings[i]; //текст i-го елементу

Memo1.Lines.Add(s); // додати до Мемо1 рядок

end;

// зміна значення позиції ScrollBar1

procedure TForm1.ScrollBar1Change(Sender: TObject);

begin

if RadioButton1.Checked **then** // якщо RadioButton1
вибрано

Edit1.Text:=IntToStr(ScrollBar1.Position); // вивести
значення позиції у Edit1

end;

// натиск на CheckBox1

procedure TForm1.CheckBox1Click(Sender: TObject);

begin

if CheckBox1.Checked **then** // якщо CheckBox1 вибрано

a:=0.5

else // інакше

```

    a:=2;
end;

// розрахунок числа Pi методом Монте-Карло
// процедура виконується кожну Timer1.Interval мілісекунду
procedure TForm1.Timer1Timer(Sender: TObject);
var x,y: integer; // координати точки
begin
    randomize; // ініціалізувати генератор випадкових чисел
    x:=Random(100); // випадкова точка x,y
    y:=Random(100);
    if x*x+y*y<=10000 then // якщо точка в колі
        begin
            k:=k+1; // кількість точок в колі
            PaintBox1.Canvas.Pixels[x,y]:=clBlack; // нарисувати
точку
        end;
        n:=n+1; // загальна кількість точок
        label2.Caption:=FloatToStr(4*k/n); // розраховане
число Pi
    end;

end. // кінець програми

```