

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БУДІВНИЦТВА І  
АРХІТЕКТУРИ**

**ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ**

Методичні вказівки  
до виконання лабораторних робіт  
для студентів спеціальності  
123 «Комп'ютерна інженерія»

Київ 2022

УДК 004.4

K52

Укладач В.В. Ключова, старший викладач

Рецензент Є.Є. Шабала, канд. техн. наук, доцент

Відповідальний за випуск Ю.І. Хлапонін, д-р техн. наук, професор

*Затверджено на засіданні кафедри кібербезпеки та комп'ютерної інженерії, протокол №3 від 08 листопада 2022 року.*

В авторській редакції.

**Візуальне програмування: Методичні вказівки до виконання**  
K52 лабораторних робіт / уклад. В.В. Ключова. – Київ: КНУБА, 2022. – 48 с.

Наведено лабораторні роботи, призначені для ознайомлення з основами програмування у середовищі Visual Basic. Розглянуто особливості створення графічного інтерфейсу та оператори мови програмування, що сприяє цілісному відтворенню процесу розробки програм засобами Visual Basic.

Призначено для студентів спеціальності 123 «Комп'ютерна інженерія».

## ЗАГАЛЬНІ ПОЛОЖЕННЯ

Метою лабораторних занять є ознайомлення з основами роботи у середовищі *Visual Basic* - інструментальною системою, що підтримує технологію візуального програмування, яка ґрунтується на об'єктно-орієнтованих принципах. Організація навчання передбачає паралельний розгляд засобів створення графічного інтерфейсу та основних операторів мови програмування, що сприяє цілісному відтворенню процесу розробки програм у даному середовищі.

Кожен студент отримує індивідуальне завдання до виконання лабораторних робіт згідно з номером варіанта, складає схему алгоритму та програму мовою програмування *Visual Basic*, виконує програму на комп'ютері з обов'язковим отриманням правильних результатів, оформлює звіт і захищає роботу перед викладачем.

Звіт з лабораторної роботи повинен містити:

- постановку задачі, вихідні дані та метод розв'язання;
- схему алгоритму (для лабораторних робіт №№ 6-10);
- вигляд форми проєкту із зазначенням використаних елементів керування;
- текст програмного коду;
- вигляд робочого вікна програми з відображенням результату.

Звіт оформлюється у друкованому вигляді на аркушах паперу формату А4.

### **Підготовка до виконання лабораторних робіт. Ознайомлення з інтерфейсом системи *Visual Basic***

Запуск *Visual Basic* здійснюється через меню **Пуск – Пуск / Программы / Microsoft Visual Studio 6.0/ Microsoft Visual Basic 6.0**.

Після запуску з'явиться діалогове вікно **New Project**. За допомогою цього вікна уточнюються наміри користувача щодо розробки нової програми (вкладка **New**), вдосконалення раніше розробленої програми (**Existing**) або вибору програми, над удосконаленням якої працювали останнім часом (**Recent**). У результаті вибору однієї з цих пропозицій потрапляємо у вікно робочого середовища *Visual Basic*. Зокрема вибір стандартного варіанта програми (**Standard EXE**) відкриє вікно середовища *Visual Basic* з чистою заготовкою форми (рис.1).

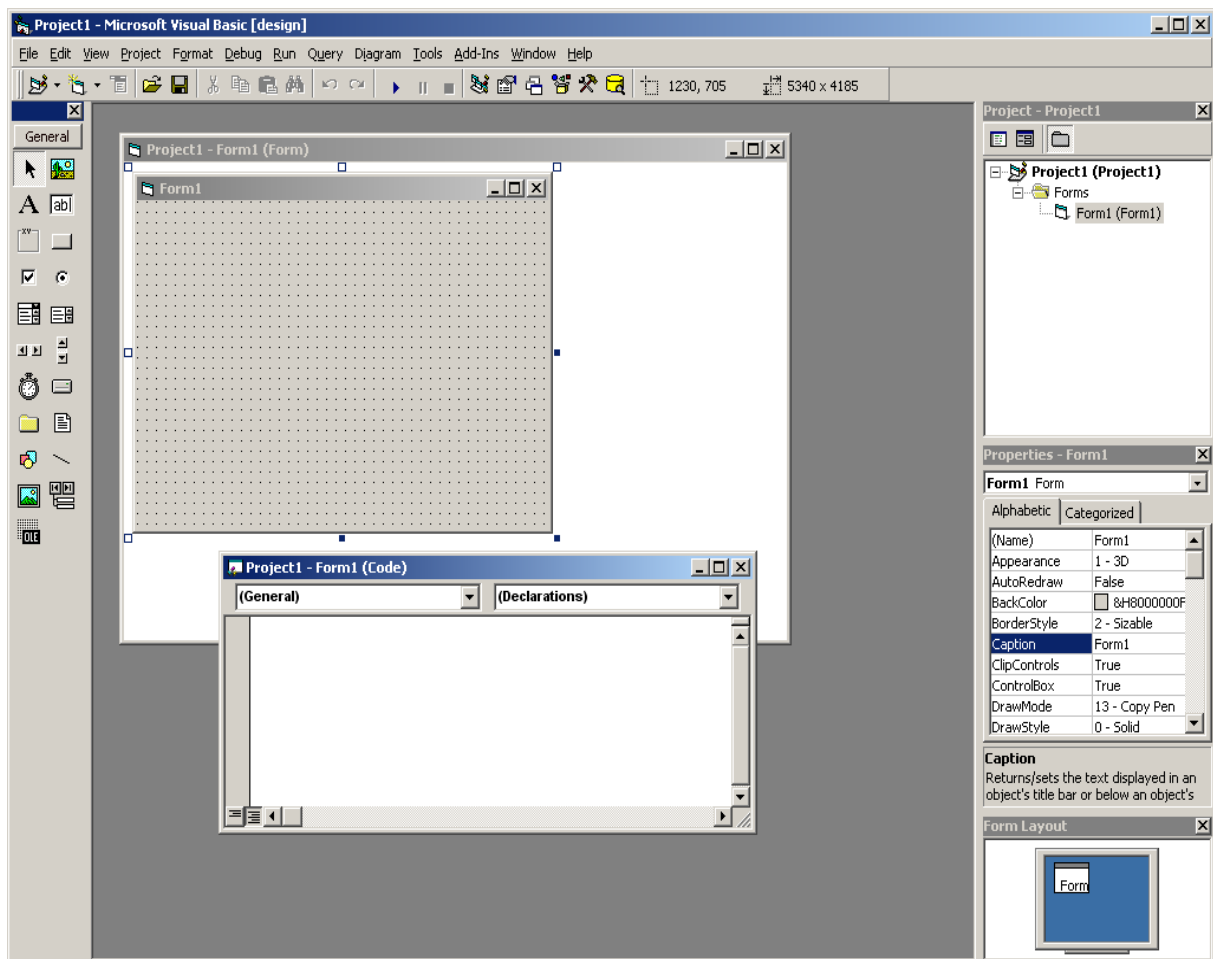


Рис. 1. Робоче середовище *Visual Basic*

У верхній частині робочого вікна *Visual Basic* розташоване головне меню, яке містить стандартний набір пунктів. Під рядком меню розміщена панель інструментів – набір кнопок із піктограмами, за допомогою яких можна отримати швидкий доступ до деяких функціональних можливостей середовища *Visual Basic*. На рис. 1 відображено лише стандартну панель інструментів (**Standard**). Інші панелі інструментів можна викликати за допомогою пункту меню **View** або контекстного меню.

Центральне місце на екрані займає вікно дизайнера форм (**Form**). Саме у цьому вікні проводиться робота із створення інтерфейсу. У вікні **Form** за допомогою елементів керування створюється макет робочого вікна програми. Цей макет називатимемо формою. Ліворуч від вікна дизайнера форм розташована панель елементів керування (**ToolBox**), за допомогою яких будується графічний інтерфейс програми.

Нижче під вікном дизайнера форм зображено вікно редактора підпрограм, тобто коду (**Code**). Це вікно відкривається за допомогою команди **View Code** або подвійного клацання мишки у полі форми.

Першим у правій колонці вікон розміщене вікно провідника проєкту (**Project Explorer**), яке відображає структурні компоненти програм, що розробляються у цей момент, і призначене для швидкого доступу до конкретної компоненти програми.

Під вікном провідника проєкту розташоване вікно властивостей (**Properties**), яке використовується для надання значень властивостям елементів керування, форми та інших об'єктів. І, нарешті, у правому нижньому кутку знаходиться вікно розташування форми (**Form Layout**), яке визначає місце форми на екрані дисплея під час виконання програми.

## *Лабораторна робота №1 Робота з проєктом. Використання основних елементів керування*

### *Хід виконання роботи:*

1. Запустіть систему *Visual Basic*. Далі виберіть режим **Standard EXE**. У вікні **Project Explorer** виділіть назву **Project1(Project1)**, а потім у вікні властивостей **Properties** в рядку (**Name**) змініть ім'я проєкту з **Project1** на **MyProject**. \

2. У вікні **Project Explorer** оберіть об'єкт **Form1(Form1)**, після чого у вікні **Properties** змініть ім'я форми з **Form1** на **MyForm1**. В рядку **Caption** змініть заголовок вікна **Form1** на "Вікно проєкту". У рядку **Height** змініть висоту форми на 5000, у рядку **Width** змініть ширину форми на 6800. Зверніть увагу на зміни, які відбулися з формою.

3. Розташуйте на формі кнопки **Command1** та **Command2**. Це робиться у вікні дизайнера форм (перемикання між вікнами дизайнера форм та редактора коду здійснюється у вікні **Project Explorer** за допомогою кнопок **View Object** та **View Code** відповідно). Елементи керування можна розташовувати на формі за допомогою подвійного клацання мишки за потрібним елементом на панелі **ToolBox** або методом перетягування. Для створення кнопок використовується елемент **CommandButton**, на піктограмі якого поміщено об'ємне зображення кнопки.

Виділіть послідовно на формі кнопки **Command1** та **Command2** і у вікні **Properties** змініть їх властивості відповідно до табл. 1.

*Таблиця 1*

<b>Властивість</b>	<b>Command1</b>	<b>Command2</b>
(Name)	CmdOk	CmdExit
Caption	Ok	Вихід
Height	375	375
Left	5600	5600
Top	750	4100
Width	1000	1000

Прослідкуйте за змінами, які відбулися на формі.

4. Аналогічно розташуйте на формі написи **Label1** та **Label2** (виберіть елементи керування **Label**, піктограма яких містить велику літеру «А») і у вікні **Properties** змініть їх властивості згідно з табл. 2.

Таблиця 2

Властивість	Label1	Label2
(Name)	LabelNr	LabelTx
Caption	Число	Текст
Height	250	250
Left	400	400
Top	300	750
Width	700	700

5. Розмістіть на формі три текстових поля. Для їх відтворення використовуються елементи типу **TextBox** з піктограмою «ab|». Змініть властивості полів **Text1** та **Text2** відповідно до табл. 3, а поля **Text3** – відповідно до табл. 4.

Таблиця 3

Властивість	Text1	Text2
(Name)	TextNbr	TextSstr
Alignment	1 – Right Justify	2 – Center
Height	300	350
Left	1200	1200
Text		
Top	300	750
Width	700	2000

Таблиця 4

Властивість	Text3
(Name)	TextMstr
Alignment	0 – Left Justify
Height	2400
Left	70
MultiLine	True
ScrollBars	2 – Vertical
Text	
Top	1500
Width	6500

Після додання перелічених елементів форма проекту повинна мати вигляд, зображений на рис. 2.

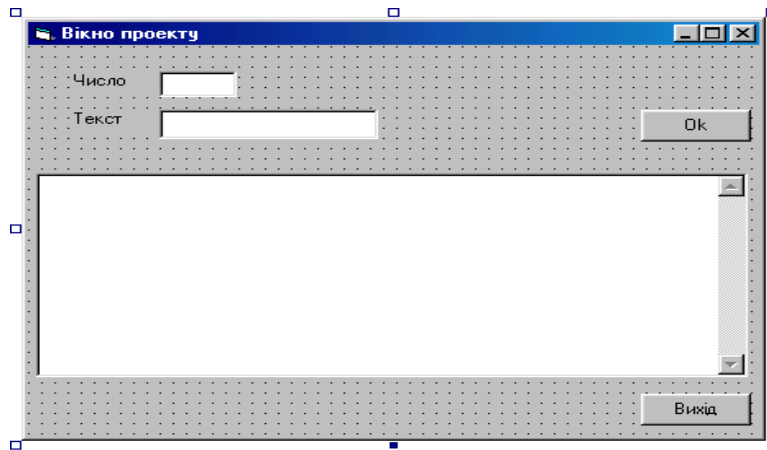


Рис. 2

6. Активізуйте вікно редактора коду командою **View Code** або натиснувши на відповідну кнопку у вікні **Project Explorer** і розгляньте його структуру. Зверніть увагу на поля зі списком об'єктів і списком подій та методів, які розташовані у вікні редактора коду зверху. Зі списку об'єктів можна вибрати ім'я об'єкта, який існує на формі, а потім для нього задати подію або метод з відповідного списку. Для кожного з об'єктів існує набір стандартних подій та методів. При застосуванні події або методу для певного об'єкта у вікні редактора автоматично з'являється відповідна заготовка базового коду відповідної процедури:

```
Private Sub <Ім'я об'єкта>_<Ім'я методу>()
```

```
Тіло процедури
```

```
End Sub
```

Тут ключові слова **Private Sub** – початок процедури, **End Sub** – кінець процедури. Між ними записується тіло процедури.

Викликати редактор коду можна також за допомогою подвійного клацання лівою клав'яшею мишки на елементі керування, для якого треба створити процедуру обробки події.

Створимо процедуру обробки події при натисканні на кнопку **Ok**. Алгоритм процедури обробки події **Click** для керуючої кнопки **CmdOk** буде полягати в тому, що при натисканні на неї дані, введені в текстові поля **TextNbr** і **TextSstr**, будуть об'єднані і виведені в третє текстове поле за допомогою текстової змінної *TxtNew*.

При написанні тексту процедури слід пам'ятати, що для швидкості вводу імені об'єкта, який існує на формі, достатньо набрати першу літеру імені, натиснути **Ctrl+J**, далі зі списку, що з'явиться праворуч від курсору, вибрати потрібне ім'я та натиснути клавішу **Tab**. Властивість об'єкта вибирається зі списку, який з'являється після введення крапки. Об'єднання даних можна виконати за допомогою символу "+", або "&".



Порядок створення даної процедури:

- зі списку об'єктів вибираємо **CmdOk** (для цього типу об'єкта подія **Click** задається автоматично);
- текстовій змінній *TxtNew* присвоюємо об'єднане значення даних з текстових полів:  
$$\text{TtxtNew} = \text{TextNbr.Text} + \text{TextSstr.Text}$$
- властивості **Text** об'єкта **TextMstr** присвоюємо значення змінної *TxtNew*:  
$$\text{TextMstr.Text} = \text{TtxtNew}$$

Самостійно створіть процедуру обробки події при натисканні на кнопку «**Вихід**». Інструкція виходу з програми – оператор **End**.

**7.** Запустіть на виконання створений проєкт. У меню **Run** виберіть пункт **Start**. Програму на виконання можна запустити також натисканням клавіші **F5** або вибором піктограми у вигляді стрілки на панелі інструментів. У випадку виявлення в програмі синтаксичної помилки її виконання буде перервано і курсор встановиться у місці можливої помилки. Якщо під час виконання виявлено неправильну роботу програми, необхідно за допомогою пункту меню **File/End** призупинити її виконання і в редакторі підпрограм внести відповідні корективи. Призупинити виконання можна також за допомогою піктограми у вигляді затемненого квадрата або кнопки «**Вихід**». Ці ж дії слід виконати після нормального завершення програми для повернення у середовище *Visual Basic*.

**8.** Доповнимо програму використанням змінних. Для цього їх потрібно попередньо оголосити.

У *Visual Basic* існує декілька способів оголошення типу даних. Можна оголосити тип за допомогою префіксів та суфіксів, які вказуються в імені змінної. Можна оголосити тип змінної за допомогою оператора:

**Ключове слово** змінна *As тип*,

де **Ключове слово** свідчить про те, що здійснюється оголошення змінної, і вказує на область її використання. *As* – службове слово, яке використовується при встановленні типу даних для змінної, що оголошується, *тип* – тип змінної.

Оголошення змінної можна здійснити в тілі процедури ключовими словами **Dim**, **Static**. При цьому вона буде доступною тільки в межах даної процедури. Щоб змінна була доступною для всіх процедур даного модуля, її необхідно оголосити в розділі **General Declarations** ключовими словами **Private** або **Dim**. Якщо в розділі **General Declarations** оголосити змінну з ключовим словом **Public**, то вона буде доступною для всіх модулів поточного проєкту.

Оголосимо змінну *n* з типом **Double**, змінну *a* з типом **String** в розділі **General Declarations**. Зі списку об'єктів вибираємо (**General**), зі списку подій вибираємо (**Declarations**). Вводимо такі інструкції:

```
Dim n As Double
```

```
Dim a As String
```

Слід пам'ятати, що в тому випадку, коли тип змінної не визначено, **Visual Basic** цій змінній за замовчанням встановлює тип **Variant**. Якщо змінна не оголошена, але використовується в процедурі, то вона є доступною тільки в даній процедурі.

Створіть на формі командну кнопку, при натисканні на яку відбувалися б такі дії:

- змінна *n* набувала значення з текстового поля **TextNbr**:  
`n = TextNbr.Text`
- змінна *m* набувала значення:  
`m = (n + 1) * 0.2`
- властивість **Text** об'єкта **TextMstr** набувала значення змінної *m*:  
`TextMstr.Text = m`

Запустіть проєкт і подивіться, як працює створена кнопка.

Додайте на форму командну кнопку, при натисканні на яку відбувалося б таке:

- змінна *a* набувала значення:  
`a = "група_"`
- змінна *m* набувала значення:  
`m = a & TextSstr.Text`
- властивість **Text** об'єкта **TextMstr** набувала значення змінної *m*:  
`TextMstr.Text = m`

Запустіть проєкт і подивіться, як працює створена кнопка.

**9.** Збережіть проєкт. Для цього у пункті меню **File** виберіть команду **Save Project**. В діалоговому вікні збереження файлу форми **Save File As** необхідно вибрати папку для збереження і вказати ім'я файлу з розширенням *.frm*, натиснути кнопку **Сохранить**. В діалоговому вікні збереження файлу проєкту **Save Project As** вказати ім'я файлу для збереження проєкту з розширенням *.vbp*, натиснути кнопку **Сохранить**. Форму **Form1** збережіть у власній папці у файлі з ім'ям *MyForm.frm*, сам проєкт збережіть у файлі з ім'ям *MyFirst.vbp*. Зверніть увагу на зміни, які відбулися у вікні **Project Explorer**.

**10.** Створіть EXE-файл поточного проєкту. Для цього необхідно в пункті меню **File** вибрати команду **Make Project.exe...**, а потім в діалоговому вікні **Make Project** вибрати папку для збереження, вказати ім'я EXE-файлу та натиснути кнопку **Ok**. Створіть EXE-файл з ім'ям *MyFirst.exe* у власній папці.

**11.** Закрийте *Visual Basic*. За допомогою програми *Мій Комп'ютер* відкрийте власну папку і завантажте створений додаток *MyFirst.exe*. Перевірте, як він працює.

## Лабораторна робота №2 Створення програм для розрахунків. Програма “Калькулятор”

Створимо додаток, який дозволяє виконувати математичні дії над двома змінними. Вибір дії буде відбуватись за допомогою елементів керування “перемикач”, які будуть розташовані в елементі керування “рамка”. Результат буде виводитись у напис “Результат” (рис. 3).

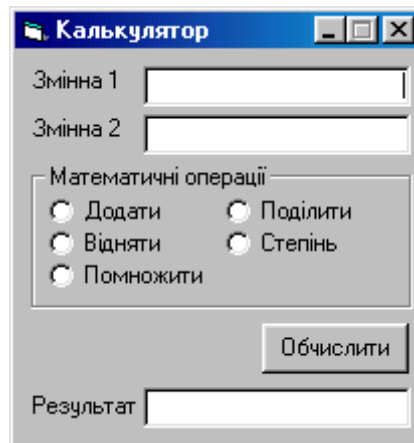


Рис. 3

### Хід виконання роботи:

1. Запускаємо *Visual Basic* і створюємо новий проєкт **Standard Exe**.
2. Для форми змінюємо властивості, наведені у табл. 5.

Таблиця 5

(Name)	FormCalc
Border Style	1 – Fixed Single
Caption	Калькулятор
Height	3300
Left	0
MaxButton	False
StartPosition	2 – CenterScreen
Top	0
Width	3100

3. Розташовуємо на формі три написи: **Label1**, **Label2**, **Label3** та змінюємо їх властивості відповідно до табл. 6.

Таблиця 6

Властивість	Label1	Label2	Label3
(Name)	LabelNr1	LabelNr2	LabelRes
Caption	Змінна 1	Змінна 2	Результат
Height	250	250	250
Left	120	120	120
Top	120	480	2550
Width	700	700	800

4. Розташовуємо на формі два текстових поля: **Text1** і **Text2** та змінюємо їх властивості відповідно до табл. 7.

Таблиця 7

Властивість	Text1	Text2
(Name)	TextV1	TextV2
Alignment	1 – Right Justify	1 – Right Justify
Height	300	300
Left	960	960
Text		
Top	120	480
Width	2000	2000

5. Розташовуємо на формі елемент **Label**, в якому буде розміщено результат обчислення, та змінюємо його властивості згідно з табл. 8.

Таблиця 8

(Name)	LabelResText
Alignment	0 – Left Justify
BackColor	Highlight Text
Border Style	1 – Fixed Single
Caption	
Height	300
Left	960
Top	2520
Width	2000

6. Розташовуємо на формі рамку **Frame1**, як показано на рис. 3, та задаємо для неї властивості, наведені у табл. 9.

Таблиця 9

(Name)	FrameOper
Caption	Математичні операції
Height	1100
Left	120
Top	840
Width	2850

7. На панелі **ToolBox** послідовно вибираємо елементи **OptionButton** та розташовуємо їх на формі в межах рамки. Відповідно їм присвоюються імена **Option1**, **Option2**, **Option3**, **Option4**, **Option5**. Змінюємо їх властивості відповідно до табл. 10.

Таблиця 10

Властивість	Option1	Option2	Option3	Option4	Option5
(Name)	OptionAdd	OptionSub	OptionMult	OptionDev	OptionDeg
Caption	Додати	Відняти	Помножити	Поділити	Степень
Height	250	250	250	250	250
Left	120	120	120	1440	1440
Top	240	480	720	240	480
Width	1210	1210	1210	1210	1210

8. Розташовуємо на формі командну кнопку **Command1** та задаємо для неї властивості, наведені у табл. 11.

Таблиця 11

(Name)	CmdCalc
Caption	Обчислити
Height	375
Left	1850
Top	2040
Width	1100

9. Збережіть проєкт у власній папці під іменем *Calculator*.

10. Створимо процедуру обробки події при натисканні на кнопку **Обчислити**. Для реалізації умовних переходів будемо використовувати структуру **If...Then...Elseif...Then...Else...End If** (можливий також скорочений варіант

оператора **If...Then... End If**). Для активізації вікна редактора коду достатньо двічі клацнути лівою клавішею мишки в полі вказаної кнопки, при цьому автоматично формується заголовок потрібної процедури. Текст процедури буде мати такий вигляд:

```
Private Sub CmdCalc_Click()  
' Оголошення змінних в тілі процедури  
Dim First As Double      'Перша змінна  
Dim Second As Double     'Друга змінна  
Dim Result As Double     'Змінна для збереження результату обчислення  
' Присвоєння змінним значень  
First = TextV1.Text  
Second = TextV2.Text  
If OptionAdd.Value = True Then ' Якщо обрано перемикач "Додати"  
Result = First + Second  
' Якщо обрано перемикач "Відняти"  
ElseIf OptionSub.Value = True Then  
    Result = First - Second  
' Якщо обрано перемикач "Помножити"  
ElseIf OptionMult.Value = True Then  
    Result = First * Second  
' Якщо обрано перемикач "Поділити"  
ElseIf OptionDev.Value = True Then  
    Result = First / Second  
' Якщо обрано перемикач "Степінь"  
ElseIf OptionDeg.Value = True Then  
    Result = First ^ Second  
End If  
LabelResText.Caption = Result  
End Sub
```

**11.** Запустіть створений проєкт і перевірте, як він працює. Закрийте вікно проєкту. Збережіть проєкт у власній папці.

### Лабораторна робота №3. Програма “Функція”

Створимо додаток, який дозволяє виконувати обчислення математичних функцій для певного аргументу. В цьому додатку вибір математичної функції буде відбуватися за допомогою елемента керування “список”. Вікно проекту зображено на рис. 4.

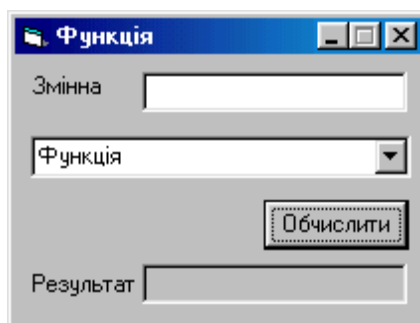


Рис. 4

#### *Хід виконання роботи:*

1. Створіть новий проект. Помістіть на форму у відповідності до рис. 4 напис з назвою “Змінна”, текстове поле з ім’ям **TextV1** для введення змінної  $a$ , командну кнопку з ім’ям **CmdCalc** та назвою “Обчислити”, напис з назвою “Результат”, напис з ім’ям **LabelResText** для виведення результату (змінна  $f$ ).

2. На панелі інструментів **ToolBox** вибираємо елемент **ComboBox**, а потім розташовуємо його на формі.

3. Для списку **Combo1**, який з’явився на формі, змінюємо властивості відповідно до табл. 12.

Таблиця 12

(Name)	ComboFun
List	(List)
Text	Функція
Height	315
Left	120
Top	600
Width	2860

Для властивості List введіть список функцій (Синус, Косинус, Тангенс, Модуль).

4. Створимо процедуру обробки події при натисканні на кнопку **Обчислити**. При написанні процедури будемо використовувати структуру з умовою **If...Then...Elseif...Then...Else...End If**. При цьому текст процедури буде мати такий вигляд:



```

Private Sub CmdCalc_Click()
' Оголошення змінних в тілі процедури
Dim a As Double
Dim f As String
' Присвоєння змінній значення
a = TextV1.Text
If ComboFun.Text = "Синус" Then           ' Якщо обрано "Синус"
f = Sin(a)
ElseIf ComboFun.Text = "Косинус" Then     'Якщо обрано "Косинус"
f = Cos(a)
ElseIf ComboFun.Text = "Тангенс" Then     'Якщо обрано "Тангенс"
f = Tan(a)
ElseIf ComboFun.Text = "Модуль" Then     'Якщо обрано "Модуль"
f = Abs(a)
Else: f = Empty
End If
LabelResText.Caption = f
End Sub

```

**5.** За допомогою властивості **BackColor** та вкладки **Palette** виділіть елементи керування різними кольорами на свій смак.

**6.** Запустіть створений проєкт і перевірте, як він працює. Закрийте вікно проєкту. Збережіть проєкт у власній папці.

## Лабораторна робота №4. Робота з меню

Створимо додаток, вікно якого містить рядок меню з пунктами **Колір** і **Розмір**. Пункт меню **Колір** містить команду **Вибір кольору**, що викликає каскадне меню з командами **Червоний**, **Синій**, **Білий**, і команду **Вихід**. Пункт меню **Розмір** містить команди **Нормальний**, **Максимальний**. Кожна команда повинна мати “гарячі клавіші” для виклику її за допомогою клавіатури. При виборі певного кольору необхідно, щоб відбувалася зміна кольору робочої області вікна програми, а при виборі розміру вікно набувало або максимального розміру, або нормального. В результаті потрібно отримати програму, вікно якої зображено на рис. 5.

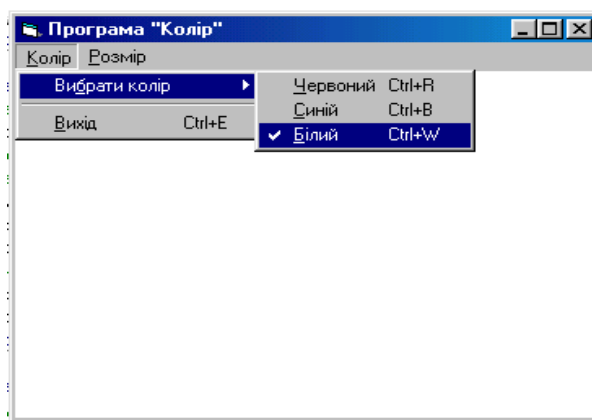


Рис. 5

### *Хід виконання роботи:*

1. Для форми проекту змініть властивості відповідно до табл. 13.

Таблиця 13

(Name)	frmColors
Caption	Програма “Колір”
Height	4300
StartPosition	2 – CenterScreen
Width	5500

2. Для створення команд меню необхідно завантажити редактор меню **Menu Editor** (командою **Tools/Menu Editor**, або комбінацією клавіш **Ctrl+E**, або відповідною кнопкою на панелі інструментів). Відкриється діалогове вікно **Menu Editor** (рис. 6).

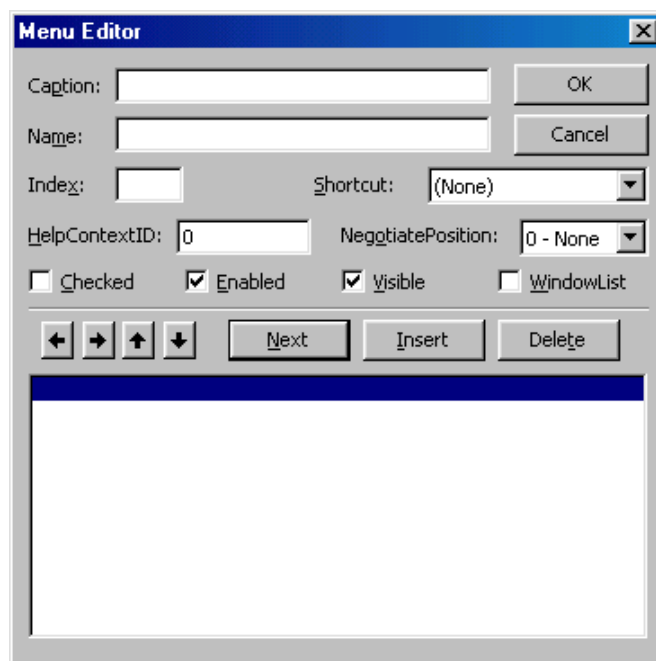


Рис. 6

В цьому вікні у поле **Caption** потрібно увести назви команд меню (властивість **Caption**) і їхні імена (властивість **Name**), щоразу натискаючи на кнопку **Next** (табл. 14).

Таблиця 14

<b>Caption</b>	<b>Name</b>
&Колір	mnuColors
&Розмір	mnuSize

У властивість **Caption** введено символ & для того, щоб в меню підкреслювалась літера, за допомогою якої можна швидко виконати пункт меню. Натисніть на кнопку **OK** в цьому вікні, а потім подивіться, що з'явилося на формі.

3. Знову активізуйте редактор меню і введіть за допомогою кнопки **Insert** назви команд меню **Колір** та комбінацію гарячих клавіш (список **Shortcut**), наведених у табл. 15.

Таблиця 15

<b>Caption</b>	<b>Name</b>	<b>Гарячі клавіши</b>
Ви&брати колір	mnuSetColor	
&Вихід	mnuExit	Ctrl+E

За допомогою кнопок зі стрілками:


 – зміна послідовності команд,
  – підпорядкування,
  – вивід з підпорядкування підпорядкуйте їх відповідному пункту меню (рис. 7).

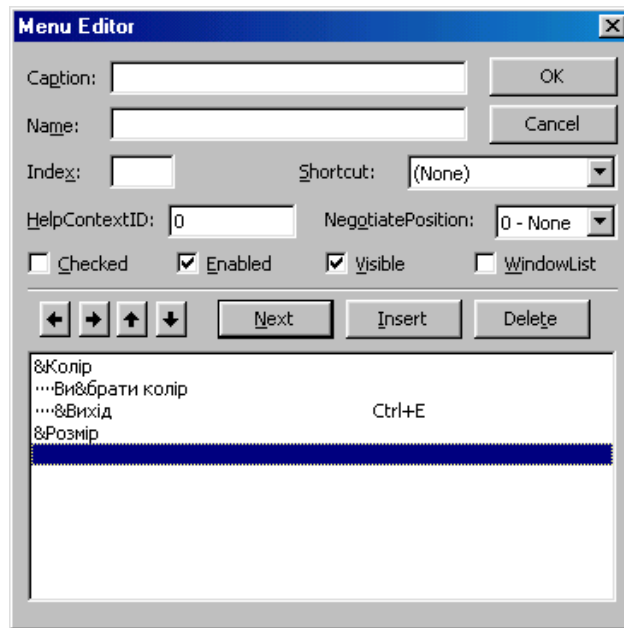


Рис. 7

Натисніть на кнопку **ОК** в цьому вікні, а потім подивіться, що відбувається на формі.

4. Створимо каскадне меню для вибору кольору за допомогою кнопки **Insert**. Активізуйте редактор меню, і введіть назви команд меню відповідно до табл. 16.

Таблиця 16

Caption	Name	Гарячі клавіші
&Червоний	mnuRed	Ctrl+R
&Синій	mnuBlue	Ctrl+B
&Білий	mnuWhite	Ctrl+W

Підпорядкуйте їх підпункту **Вибрати колір**, а також додайте лінійно-сепаратор для відокремлення частин (рис. 8).

Натисніть на кнопку **ОК** в цьому вікні, а потім подивіться, що відбувається на формі.

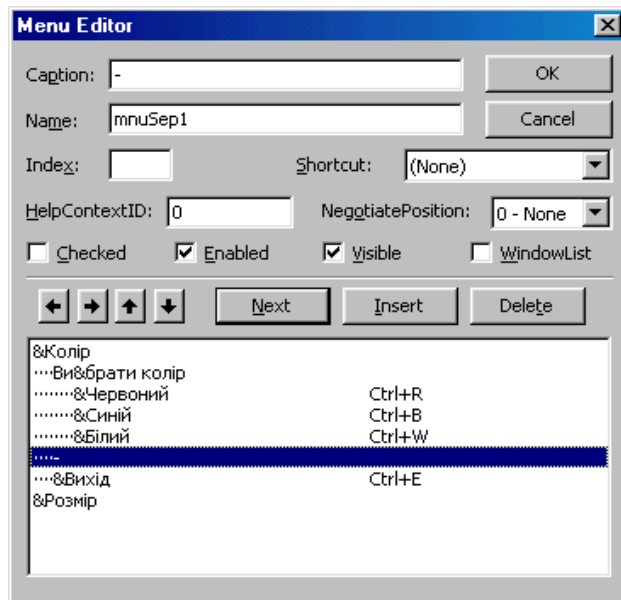


Рис. 8

5. Активізуйте редактор меню, введіть самостійно назви команд меню **Розмір** та комбінацію гарячих клавіш, наведених у табл. 17.

Таблиця 17

Caption	Name	Гарячі клавіши
&Нормальний	mnuNorm	Ctrl+N
&Максимальний	mnuMax	Ctrl+M

Підпорядкуйте їх відповідному пункту меню. Також додайте лінію-сепаратор. В результаті вікно **Menu Editor** повинно мати вигляд, зображений на рис. 9.

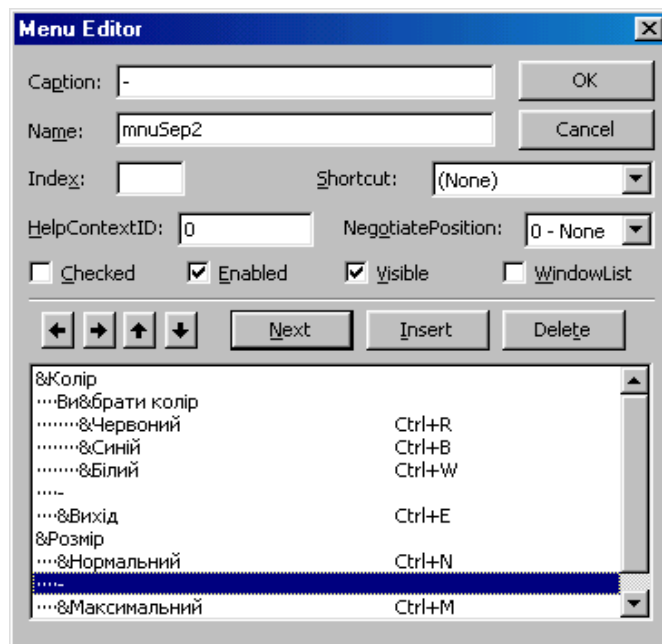


Рис. 9

Натисніть на кнопку **ОК** в цьому вікні, а потім подивіться, що відбувається на формі.

6. Створимо процедури обробки подій при виборі команд у пунктах меню:

```
Private Sub mnuRed_Click()  
' Вибрати колір фону вікна - червоний  
frmColors.BackColor = vbRed  
' Встановлення прапорця  
mnuRed.Checked = True  
mnuBlue.Checked = False  
mnuWhite.Checked = False  
End Sub  
Private Sub mnuBlue_Click()  
' Вибрати колір фону вікна - синій  
frmColors.BackColor = vbBlue  
' Встановлення прапорця  
mnuRed.Checked = False  
mnuBlue.Checked = True  
mnuWhite.Checked = False  
End Sub  
Private Sub mnuWhite_Click()  
' Вибрати колір фону вікна - білий  
frmColors.BackColor = vbWhite  
' Встановлення прапорця  
mnuRed.Checked = False  
mnuBlue.Checked = False  
mnuWhite.Checked = True  
End Sub  
Private Sub mnuNorm_Click()  
' Встановити нормальний розмір вікна  
frmColors.WindowState = 0  
' Вимкнути пункт "Нормальний"  
mnuNorm.Enabled = False  
' Ввімкнути пункт "Максимальний"  
mnuMax.Enabled = True  
End Sub  
Private Sub mnuMax_Click()  
' Встановити максимальний розмір вікна
```

```
frmColors.WindowState = 2
' Вимкнути пункт "Максимальний"
mnuMax.Enabled = False
' Ввімкнути пункт "Нормальний"
mnuNorm.Enabled = True
End Sub
Private Sub mnuExit_Click()
End
End Sub
```

**7.** Запустіть проєкт і перевірте, як він працює. Закрийте вікно проєкту. Збережіть проєкт у власній папці.

## Лабораторна робота №5. Робота з графікою

Розробимо додаток, який дозволяє створювати малюнки на формі. В цьому додатку буде здійснюватися малювання довільної лінії, прямої лінії методом **Line**, кола методом **Circle** за допомогою лівої клавіші миші. Вибір фігури буде здійснюватися з пункту меню **Малювання**, яке складається з команд: **Довільна лінія**, **Пряма лінія**, **Коло**. Меню **Малювання** також повинно містити команди **Очистити форму** і **Вихід** (рис. 10). При малюванні всіх фігур курсор миші повинен мати вигляд “+”. Довільна лінія повинна малюватися чорним кольором при переміщенні миші з натиснутою лівою клавішею, пряма лінія – червоним кольором, коло – зеленим.

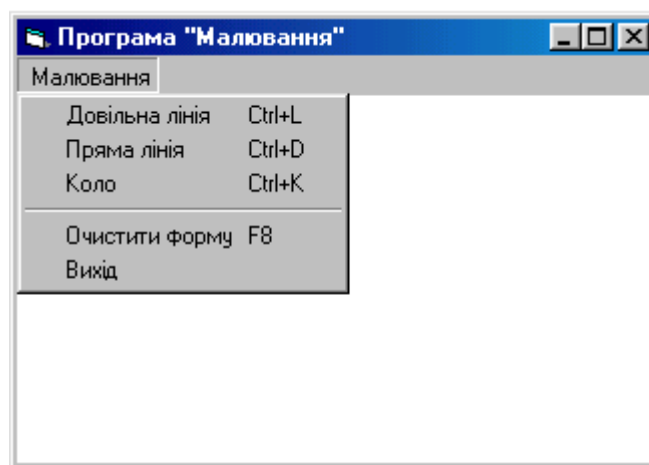


Рис. 10

### Хід виконання роботи:

1. Самостійно створіть меню відповідно до табл. 18.

Таблиця 18

Caption	Name	Гарячі клавіши
Малювання	mnuDraw	
Довільна лінія	mnuLine	Ctrl+L
Пряма лінія	mnuDrLine	Ctrl+D
Коло	mnuCircle	Ctrl+K
Очистити форму	mnuCls	F8
Вихід	mnuExt	

2. Для відстеження того, щоб усі змінні були оголошені явно, у вікні редактора коду в розділі **General Declarations** введемо такий оператор:

Option Explicit

3. Для вибору типу фігури введемо змінну *LineType*, яку оголосимо в розділі **General Declarations** з типом **Integer**:



Dim LineType As Integer

Для довільної лінії змінна *LineType* буде набувати значення 1, для прямої лінії – 2, для кола – 3.

4. Створимо процедуру обробки події вибору фігури **Довільна лінія** з пункту меню **Малювання**. Для цього переходимо в редактор коду, зі списку об'єктів, що розташований зліва в верхній частині вікна редактора, вибираємо **mnuLine** і описуємо процедуру обробки події **Click**:

```
Private Sub mnuLine_Click()  
mnuLine.Checked = True      ' Встановлення відмітки в меню  
                             ' вибраної команди “Довільна лінія”  
mnuDrLine.Checked = False  ' Скасування відмітки в меню вибраної  
                             ' команди “Пряма лінія”  
mnuCircle.Checked = False  ' Скасування відмітки в меню вибраної  
                             ' команди “Коло”  
LineType = 1                ' Вибір Довільної лінії  
End Sub
```

5. Збережіть проєкт.

6. Самостійно створіть процедури обробки події вибору фігур **Пряма лінія** і **Коло** з пункту меню **Малювання**.

7. Далі почнемо створювати процедури для малювання об'єктів. Для малювання кожної фігури потрібно натисканням лівої клавіші миші визначати початкову точку з координатами *CurrentX*, *CurrentY*. При цьому виникає подія **MouseDown**, а також змінній *Button* присвоюється значення 1, змінним *X*, *Y* присвоюються поточні значення координат курсору миші. В процедурі обробки події **MouseDown** необхідно, щоб змінні *CurrentX* і *CurrentY* набували поточних значень *X*, *Y* як початкових. Властивість **MousePointer** визначатиме вигляд курсору. Процедура обробки цієї події буде мати вигляд:

```
Private Sub Form_MouseDown(Button As Integer, _  
Shift As Integer, X As Single, Y As Single)  
' Збереження координат точки, де була натиснута кнопка миші  
' у змінних CurrentX і CurrentY  
CurrentX = X  
CurrentY = Y  
' Зміна вигляду курсору на хрестик  
MousePointer = 2  
End Sub
```

8. Збережіть проєкт. Запустіть його і перевірте, як він працює.

9. При пересуванні миші по формі повинно відбуватися малювання довільної лінії. Виникає подія **MouseMove**. Малювання лінії в тілі процедури відбувається методом **Line**. Синтаксис методу **Line** має такий вигляд:

*object.Line* (X1, Y1) – (X2, Y2), *color*, BF

*object* – назва об'єкта (необов'язковий параметр, у нашому випадку це форма, на якій ми малюємо);

(X1, Y1) – координати початкової точки;

(X2, Y2) – координати кінцевої точки, до якої буде проведена лінія;

*color* – колір лінії (необов'язковий параметр);

B – малювання прямокутника;

F – заповнення кольором прямокутника.

Створимо процедуру обробки події **MouseMove**:

```
Private Sub Form_MouseMove(Button As Integer, _  
Shift As Integer, X As Single, Y As Single)  
If LineType = 1 And Button = 1 Then  
Line -(X, Y), vbBlack  
End If  
End Sub
```

10. Збережіть проєкт. Запустіть його і перевірте, як він працює.

11. Малювання прямої лінії і кола повинно відбуватися при відпусканні клавіші миші (подія **MouseUp**). Малювання кола в тілі процедури виконується методом **Circle**. Синтаксис методу **Circle** має такий вигляд:

*object.Circle* (X, Y), radius, *color*

*object* – необов'язковий параметр;

(X, Y) – координати центра кола;

radius – радіус кола;

*color* – колір лінії кола.

Створимо процедуру обробки події **MouseUp**:

```
Private Sub Form_MouseUp(Button As Integer, _  
Shift As Integer, X As Single, Y As Single)  
' Малювання прямої лінії  
If LineType = 2 And Button = 1 Then
```

```

Line -(X, Y), vbRed
End If
' Малювання кола
If LineType = 3 And Button = 1 Then
' Обчислення радіуса
Dim R As Single
R = Sqr((X - CurrentX) ^ 2 + (Y - CurrentY) ^ 2)
Circle (CurrentX, CurrentY), R, vbGreen
End If
' Зміна вигляду курсору на стрілку
MousePointer = 0
End Sub

```

**12.** Збережіть проєкт. Запустіть його і перевірте, як він працює.

**13.** Самостійно додайте команду меню і створіть процедуру для малювання зафарбованого прямокутника.

**14.** Створимо процедуру обробки подій при виборі команди **Очистити форму**:

```

' Очищення форми
Private Sub mnuCls_Click()
Cls
End Sub

```

**15.** Створимо процедуру обробки подій при виборі команди **Вихід**:

```

' Вихід
Private Sub mnuExt_Click()
End
End Sub

```

**16.** Збережіть проєкт. Запустіть його і перевірте, як він працює.

**Лабораторна робота №6. Робота з підпрограмами, діалоговими вікнами, організація циклів. Задача «Таблювання функції»**

Створимо додаток, який дозволяє обчислювати значення функції  $y=f(x)$  та будувати її графік на інтервалі значень аргументу  $x$  від початкового значення до кінцевого з кроком  $dx$ . При завантаженні додатка повинно з'являтися діалогове вікно з привітанням. Приклад робочого вікна зображено на рис. 11. Варіанти завдань наведено у табл. 19. Початкове значення аргументу  $x$  повинно вводитися в текстове поле “Ліва межа”, кінцеве значення – в текстове поле “Права межа”, крок – в текстове поле “Крок”. Результат табулювання функції буде виводитись в багаторядкове текстове поле. Побудова графіка функції буде відбуватися в графічному полі. Вікно проекту повинно мати рядок меню з пунктом **Функція**, який містить в собі команди: **Таблювати**, **Побудувати графік**, **Очистити**, **Вихід**.

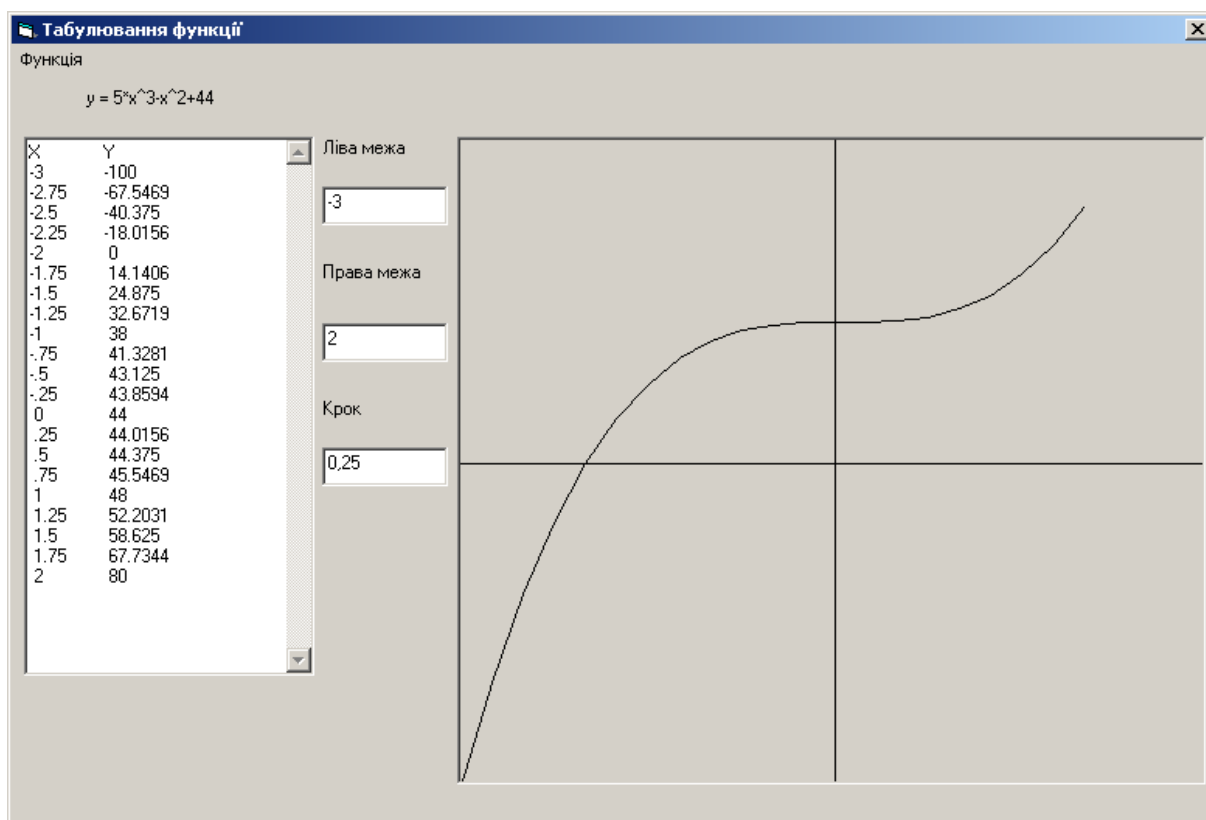


Рис. 11

Таблиця 19

№ пор.	Функція	Інтервал		Крок $dx$
		$x_p$	$x_k$	
1	$y = 0,1x^2 - x \ln x$	1	2	0,05
2	$y = x^3 - 1,473x^2 - 5,738x + 6,763$	-3	0	0,15
3	$y = x^2 - x - 5$	2	4	0,1
4	$y = x^3 - 0,39x^2 - 10,5x + 11$	-5	-2	0,15
5	$y = 4x^3 - 12,3x^2 - x + 16,2$	-2	0	0,1
6	$y = x^4 - 26x^3 + 131x^2 - 226x + 110$	-1	1	0,1
7	$y = x^3 - 2x^2 - 3x + 9$	-2	1	0,15
8	$y = x^5 + 8x^4 + 17x^3 - 8x^2 - 14x + 20$	-2	1	0,15
9	$y = 2 \cos x - 4,5x + 0,76$	0	1	0,05
10	$y = 2 \sin x + 6,53x - 6,7$	0	1	0,05
11	$y = 3,5e^x - 8,65x - 2,43$	1	3	0,1
12	$y = 2,5\sqrt{ 6x+1 } + 4x - 3,6$	-2	2	0,2
13	$y = 0,35 \cdot 10^{x+1,6} - 104,7x - 183$	1,2	2	0,04
14	$y = \operatorname{tg}(x+1,6) + 0,5x + 0,02$	1	1,54	0,027
15	$y = 6x^7 + 1,54x^3 + 6,56$	-2	0	0,1
16	$y = 0,18e^{x+3,6} - 5,7(x+1) - 8,3$	0	3	0,15
17	$y = x^4 + 2,4x^3 - 2,85x + 0,13$	0,5	2	0,075
18	$y = (x^{x^2} + 6x + 2) / (x \cdot 10^5) - 1,5x - 12,3$	2	3,6	0,08
19	$y = 2x^5 - 7,6x^3 + 4,3x - 3,4$	1	3	0,1
20	$y = 35,4e^{x-1} + 3x - 4,72$	-1	1	0,1
21	$y = 5x^2 - 6x - 8$	-1	1	0,1
22	$y = x^3 - 13,5x^2 + 27$	-1	2	0,15
23	$y = 3 \sin x + 4,5x - 19,6$	0	6	0,3
24	$y = \cos x^3 - 5x + 10,15$	0	3	0,15
25	$y = \sin 5x - 6,5x + 9,88$	-1	2	0,15

**Хід виконання роботи:**

1. Для форми проєкту змініть властивості, наведені у табл. 20.

Таблиця 20

(Name)	FrmFunction
BorderStyle	1 – Fixed Single
Caption	Табуювання функції
Height	7600
MaxBatton	False
MinBatton	False
StartUpPosition	2 – CenterScreen
Width	11700

2. Помістіть на форму багаторядкове текстове поле для виведення результатів табулювання функції (значень  $x$  та  $y$ ) з властивостями, наведеними у табл. 21.

Таблиця 21

(Name)	TextFun
MultiLine	True
Text	
ScrollBars	2 - Vertical
Height	5200
Left	120
Top	600
Width	2800

3. Самостійно помістіть на форму три текстових поля для введення лівої та правої границь значення аргументу, а також кроку з іменами відповідно **TextLeft**, **TextRight**, **TextStep** так, як показано на рис. 11.

4. Помістіть на форму графічне поле **PictureBox** для побудови графіка функції з властивостями, наведеними у табл. 22.

Таблиця 22

(Name)	PictureGraph
BorderStyle	1 – Fixed Single
Height	6250
Left	4300
Top	600
Width	7230

5. Самостійно помістіть на форму написи зі змістом:  $y=f(x)$ , Графік функції, Ліва межа, Права межа, Крок (замість  $f(x)$  вкажіть формулу своєї функції відповідно до синтаксису мови **Visual Basic**). Розташуйте їх так, як показано на рис. 11.

6. Створіть рядок меню з пунктом **Функція**, який має ім'я **mnuFun**. В ньому повинні знаходитись команди меню з властивостями, заданими у табл. 23.

Таблиця 23

Caption	Name	Гарячі клавіши
Табулювати	mnuTab	Ctrl+F
Побудувати графік	MnuGraph	Ctrl+G
Очистити	MnuClear	F8
Вихід	MnuEnd	

7. Відкрийте вікно редактора коду та в розділі **General Declarations** оголосіть змінні:

```
Dim x As Double
Dim y As Double
Dim xn As Double
Dim xk As Double
Dim dx As Double
Dim i As Integer
Dim NewLine As String
Dim STab As String
Dim ymax As Double
```

8. Створіть підпрограму для зчитування вихідних даних з текстових полів:

```
Sub Coordinats()
  xn = TextLeft.Text
  xk = TextRight.Text
  dx = TextStep.Text
End Sub
```

9. Запишіть підпрограму для обчислення функції  $y=f(x)$  згідно зі своїм варіантом. Для прикладу візьмемо функцію  $y = 5x^3 - x^2 + 44$ .

```
' Підпрограма обчислення функції
Sub Tabuls()
  y = 5 * x ^ 3 - x ^ 2 + 44
End Sub
```

10. Створимо процедуру, яка дозволяє організовувати обчислення заданої функції для значень  $x$  з кроком  $dx$  в межах від  $xn$  до  $xk$ , а потім значення  $x$  і  $y$  виводити в текстове поле **TextFun**. Для цього використаємо оператор циклу **For...Next**, формат якого має вигляд:

**For** лічильник = п.з. **To** к.з. **Step** крок

Тіло циклу

**Next** лічильник

**To** –ключове слово, що вказує на кінцеве значення змінної;

**Step** –ключове слово, що вказує на *крок*;

лічильник – змінна, якій надаються значення від початкового до кінцевого з кроком *крок* під час виконання циклу;

п.з. –початкове значення змінної;

к.з. –кінцеве значення змінної.

Уведіть текст процедури:

```
Private Sub mnuTab_Click()  
NewLine = Chr(13) + Chr(10)           ' Enter  
STab = Chr(9)                          ' Tab  
TextFun.Text = "X" + STab + "Y" + NewLine      'Шапка таблиці  
Call Coordinats ' Виклик підпрограми введення початкових значень  
For x = xn To xk+0.01 Step dx  
Call Tabuls          ' Виклик підпрограми обчислення функції  
TextFun.Text = TextFun.Text + Str(Format(x, "0.00")) + STab + _  
Str(Format(y, "0.0000")) + NewLine  
Next x  
End Sub
```

**11.** Створимо процедуру, яка дозволяє визначати масштаб побудови графіка, організувати обчислення заданої функції для значень  $x$  з кроком  $dx$  в межах від  $x_n$  до  $x_k$  та здійснювати побудову графіка:

```
Private Sub mnuGraph_Click()  
'Побудова осей X та Y  
CoordY = PictureGraph.Height / 2  
CoordX = PictureGraph.Width / 2  
PictureGraph.Line (0, CoordY)-(PictureGraph.Width, CoordY)  
PictureGraph.Line (CoordX, 0)-(CoordX, PictureGraph.Height)  
Call Coordinats 'Виклик підпрограми введення початкових значень  
' Визначення масштабу по осі X  
If Abs(xn) > Abs(xk) Then  
    sclx = 0.5 * PictureGraph.Width / Abs(xn)  
    Else: sclx = 0.5 * PictureGraph.Width / Abs(xk)  
End If  
' Визначення масштабу по осі Y  
x = xn
```



```

Call Tabuls
ymax = Abs(y)
For x = xn + dx To xk Step dx
    Call Tabuls
    If Abs(y) > ymax Then ymax = Abs(y)
Next
sclx = 0.5 * PictureGraph.Height / ymax
' Побудова графіка
For x = xn To xk + 0.1 Step dx
    Call Tabuls
    If x = xn Then xpred = x: ypred = y
    PictureGraph.Line ((xpred * sclx + CoordX), (-ypred * sclx + CoordY)) _
        -((x * sclx + CoordX), (-y * sclx + CoordY))
    xpred = x
    ypred = y
Next x
End Sub

```

**12.** Створимо процедуру для очищення текстового поля з результатом і графічного поля:

```

Private Sub mnuClear_Click()
    TextFun.Text = Empty           ' Очищення текстового поля
    PictureGraph.Cls              ' Очищення графічного поля
End Sub

```

**13.** Створимо процедуру, яка дозволяє виводити діалогове вікно з привітанням при завантаженні додатка. Для цього використаємо функцію **MsgBox**. Загальний вигляд функції **MsgBox** такий:

**MsgBox**(Message, Button, Title),

де Message – повідомлення на екрані, Button – одна з констант *Visual Basic*, що визначає набір кнопок та значок у вікні повідомлення **MsgBox** (наприклад, константа vbOKOnly – наявність у вікні повідомлення кнопки **ОК**), Title – назва вікна повідомлення.

Уведіть текст процедури:

```

Private Sub Form_Load()
Dim m
m = MsgBox("Вас вітає програма табулювання та побудови графіка _
    функції y = 5 * x ^ 3 - x ^ 2 + 44", vbOKOnly)

```

End Sub

В результаті виконання цієї процедури з'являється діалогове вікно (рис. 12).

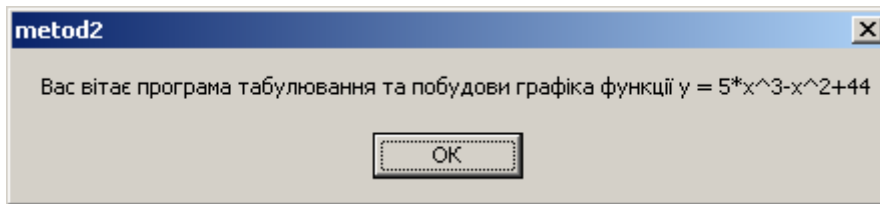


Рис. 12

14. Зробіть кольорове оформлення створеного вікна додатка.
15. Збережіть проєкт та перевірте, як він працює.

## Лабораторні роботи для самостійного виконання

### Лабораторна робота №7. Чисельне інтегрування

Зміст завдання: необхідно створити проєкт, який би дозволяв обчислювати визначений інтеграл наближеним методом за однією з нижченаведених формул:

$$\int_a^b f(x)dx \approx h \sum_{i=1}^n f(a + ih) \text{ – формула правих прямокутників;}$$

$$\int_a^b f(x)dx \approx h \sum_{i=1}^n f(a + (i - 1)h) \text{ – формула лівих прямокутників;}$$

$$\int_a^b f(x)dx \approx h \sum_{i=1}^n f(a + h/2 + (i - 1)h) \text{ – формула середніх прямокутників;}$$

$$\int_a^b f(x)dx \approx \frac{h}{2} \left[ f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(a + ih) \right] \text{ – формула трапецій,}$$

$$\text{де } h = \frac{b - a}{n}.$$

Приблизний вигляд робочого вікна проєкту зображено на рис. 13. Приклад алгоритму для обчислення значення визначеного інтеграла за формулою лівих прямокутників показано на рис. 14.

Варіанти завдань наведено у табл. 24.

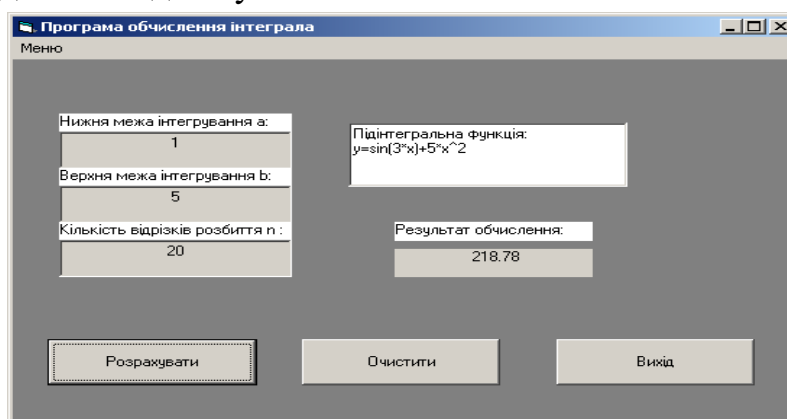


Рис. 13

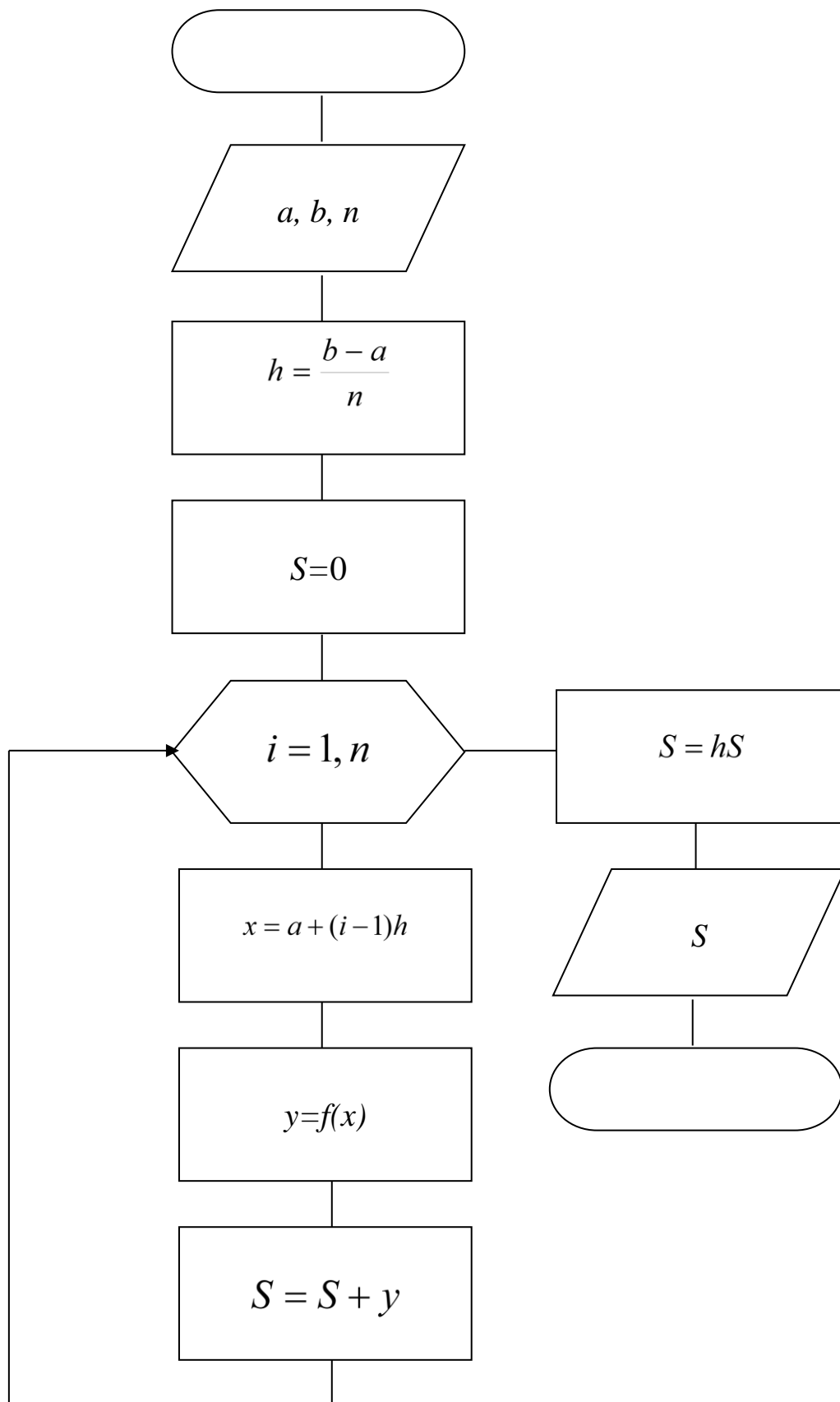


Рис. 14. Схема алгоритму обчислення визначеного інтеграла за формулою лівих прямокутників

## Варіанти завдань

№ пор.	Підінтегральна функція	Проміжок інтегрування		Кількість відрізків розбиття, $n$	Формула для обчислення
		$a$	$b$		
1	2	3	4	5	6
1	$(1 + e^{-x}) / (2 + \sin x)$	0	1	60	лівих прямокутників
2	$xtg^2(5 - xe^{-x^2})$	1	4	20	середніх прямокутників
3	$5 \sin^3 \operatorname{arctg} x$	0,5	1,5	70	правих прямокутників
4	$((x^2 - 5) / \sqrt[3]{x^2 + 1}) e^{-x}$	1	4	30	трапецій
5	$e^{\pi x} \sqrt{\sin x + 3}$	8	22,5	100	лівих прямокутників
6	$1 / (x + \sin(2x + \sqrt{x^2 + 1}))$	1	1,4	40	середніх прямокутників
7	$6 \cos \sqrt{ \cos x }$	7	12	110	правих прямокутників
8	$tg(x + x^4) / (x^4 + 3x^3)$	1	2	30	трапецій
9	$\sqrt{ \sin  4 \sin x }$	12	55	80	лівих прямокутників
10	$4 \sqrt[4]{ x^2 + 7,5x - 8 }$	1	2	30	середніх прямокутників
11	$e^x (e^{-x} + 0,2 \sin e^{-x})$	79	80	20	правих прямокутників
12	$x^2 + \ln(x + 1) e^{-x\sqrt{x}}$	0,3	2,5	30	трапецій
13	$\cos(5,2 + 6,3\sqrt{x^2 + x - 1})$	3	23	60	лівих прямокутників
14	$(e^{-x} + xe^{-2x} + 2) \sin 3x$	1,5	2,5	30	середніх прямокутників
15	$\sin^3(\cos x) + \sin(x \cos x)$	2,7	2,85	50	правих прямокутників

1	2	3	4	5	6
16	$\sin(x/\sqrt{1+\lg(x+5)})$	1	3,5	30	трапецій
17	$e^{\cos x} + 5,47 \arccos(0,01x)$	19,3	44,5	30	лівих прямокутників
18	$1/\sqrt[3]{8x^3 + 2x \sin x}$	1	2	40	середніх прямокутників
19	$\sin x + \sqrt{x^2 + \cos(55,3x)}$	-8,5	-6	100	правих прямокутників
20	$x^2 e^{-2x} \sin^3(x\sqrt{x})$	1	2	40	трапецій
21	$e^x(4 \sin x + 10)$	2	4	120	лівих прямокутників
22	$\sin(x + \ln(x/(x^3 + 5)))^2$	1	1,5	50	середніх прямокутників
23	$\sqrt{2x + 3e^x}$	1	10	80	правих прямокутників
24	$\sin 3x + 2 \cos(x^5 + 2x)$	1	2	40	трапецій
25	$4 \cos \sqrt{e^x + 1}$	3	19	90	лівих прямокутників

## Лабораторна робота №8. Визначення кореня алгебраїчного рівняння

Зміст завдання: розробити проєкт числового розв'язування алгебраїчного рівняння методом половинного ділення (дихотомії).

### Постановка задачі

Дано:  $f(x) = 0$  – рівняння;

$[a, b]$  – інтервал, на якому повинен бути визначений корінь рівняння;

$\varepsilon = 0,1 \cdot 10^{-3}$  – точність розв'язку задачі.

Знайти  $x$ , при якому  $|f(x)| \leq \varepsilon$ .

На початку програми необхідно перевірити, чи існує на заданому інтервалі корінь. Для цього треба обчислити значення функції на кінцях інтервалу. Якщо ці значення мають різні знаки, корінь існує і його визначення треба здійснювати за наведеним нижче алгоритмом. У протилежному випадку необхідно вивести повідомлення про те, що кореня на заданому інтервалі немає.

Алгоритм методу дихотомії складається з таких дій:

1. Визначити середню точку заданого інтервалу  $x = \frac{a+b}{2}$ .

2. Якщо  $f(x)$  і  $f(a)$  мають різні знаки, то значення кореня лежить на відрізку  $[a, x]$ , в іншому випадку – на відрізку  $[b, x]$ . Звуження діапазону пошуку, тобто використання значення  $x$  як правої або лівої межі інтервалу знаходження кореня ( $b = x$  або  $a = x$ ).

3. Якщо значення функції  $f(x)$  за абсолютною величиною менше заданої точності  $\varepsilon$ , то  $x$  можна вважати коренем рівняння. Перейти до пункту 4. В іншому випадку перейти до пункту 1.

4. Вивести значення функції та її аргументу. Завершення алгоритму.

Вказівка. Формулу для визначення  $f(x)$ , початкове та кінцеве значення інтервалу  $[a, b]$  слід взяти з табл. 19 ( $a=xp$ ,  $b=xk$ ). Для організації циклу бажано використати конструкцію **Do...Loop**. Доповніть алгоритм підрахунком кількості кроків виконання циклу. Порівняйте результат з графічним розв'язком рівняння, отриманим у лабораторній роботі №6.

У робочому вікні проєкту відобразіть формулу алгебраїчного рівняння, інтервал пошуку кореня, значення кореня та функції в цій точці, кількість

кроків виконання циклу, кнопку «Вихід», а також передбачте виведення повідомлення про те, що кореня на заданому інтервалі немає.



## Лабораторна робота №9. Сортування масиву

Зміст завдання: розробіть проєкт для упорядкування масиву з 15 цілих чисел за зростанням одним з чотирьох методів, алгоритми яких описані нижче. Варіанти завдань наведені у табл. 25.

Вказівка. Введення масиву чисел необхідно здійснювати у масив текстових полів, який створюється шляхом копіювання елемента **TextBox** необхідну кількість разів. При цьому система **Visual Basic** автоматично присвоює кожному елементу чергове значення індексу, починаючи з нульового. Зчитування інформації має відбуватися у масив даних, для оголошення якого використовується оператор:

Dim k(15) As Integer

Для виводу впорядкованого масиву можна використати багаторядкове текстове поле.

Таблиця 25

### Варіанти завдань

Номери варіантів	Метод сортування
1,5,9,13,17,21,25	простої вставки
2,6,10,14,18,22	заміни
3,7,11,15,19,23	простого вибору
4,8,12,16,20,24	бульбашки

### Сортування методом простої вставки

Масив складається з елементів  $k_1, k_2, \dots, k_j, \dots, k_n$ .

Нехай елементи  $k_1, k_2, k_3, \dots, k_{j-1}$  розміщені так, що  $k_1 \leq k_2 \leq \dots \leq k_{j-1}$ . Елемент  $k_j$  порівнюється по черзі з  $k_{j-1}, k_{j-2}$  і так далі доти, поки не знайдуться такі  $k_i$  і  $k_{i+1}$ , що  $k_i \leq k_j \leq k_{i+1}$ . Тоді елементи  $k_{i+1}, k_{i+2}, \dots, k_{j-2}, k_{j-1}$  слід посунути праворуч на одне місце і елемент  $k_j$  розмістити у позиції  $k_{i+1}$ .

Алгоритм методу має вигляд:

1. Початкова установка  $j=2$ .
2. Визначення  $i=j-1, k=k_j$ .
3. Порівняння  $k$  і  $k_i$ . Якщо  $k \geq k_i$  – перейти до п. 6, у протилежному випадку – до п. 4.
4. Визначення  $k_{i+1}=k_i, i=i-1$ .
5. Порівняння. Якщо  $i > 0$  – перейти до п.3, інакше – до п. 6.
6. Розмістити  $k$  на місці  $k_{i+1}$ .

7. Зміна  $j$  за формулою  $j=j+1$ .
8. Порівняння  $j \leq n$ . Якщо  $j \leq n$  – перейти до п. 2, якщо  $j \geq n$  – до п. 9.
9. Кінець сортування.

#### *Сортування методом заміни*

У масиві  $\{k_j\}$ ,  $j = \overline{1, n}$  визначаються  $j$  і  $j+1$ -й елементи і порівнюються між собою ( $j$  змінюється від 1 до  $n-1$ ). Якщо  $k_j > k_{j+1}$ , відбувається обмін елементами, а потім перехід до іншої пари  $k_{j+1}$  і  $k_{j+2}$  і т.д. При цьому більші елементи просуваються праворуч. Найбільший елемент займає положення елемента  $k_n$ . Потім знову починається перегляд з пари елементів  $k_j$  і  $k_{j+1}$  ( $j$  змінюється від 1 до  $n-2$ ). При другому перегляді масиву найбільший елемент займає положення елемента  $k_{n-1}$ . При багаторазовому виконанні процесу відповідні елементи розміщуються у позиціях  $n-3$ ,  $n-4$  і т.д., так що по закінченні всі елементи будуть упорядковані за зростанням. У загальному випадку кількість переглядів повинна бути  $n-1$ , але можливі варіанти, коли упорядкування масиву відбудеться на  $m$ -му кроці

$$1 < m < n-1.$$

При цьому слід зупинити процедуру сортування. З цією метою використовується ознака  $P$ , що приймає значення 1, якщо перестановка елементів відбулася при  $m$ -му перегляді, і значення 0 у протилежному випадку.

Алгоритм цього методу має вигляд:

1. Початкова установка  $j=1$ ,  $n=n-1$ ,  $P=0$ .
2. Порівняння  $k_j$  і  $k_{j+1}$ . Якщо  $k_j > k_{j+1}$  – перейти до п. 3, у протилежному випадку – до п. 4.
3. Обмін  $k_j \leftrightarrow k_{j+1}$ , присвоєння ознаці  $P=1$ , перехід до п. 4.
4. Зміна  $j$  за формулою  $j=j+1$ .
5. Порівняння  $j$  і  $n$ . Якщо  $j \leq n$  – перейти до п. 2, у протилежному випадку – до п. 6.
6. Перевірка. Якщо  $P=0$  – перейти до п. 7, інакше – до п. 1.
7. Кінець сортування.

#### *Сортування методом простого вибору*

Елементи перерозміщуються на тому самому місці. Вибране значення можна записати у позицію, де знаходиться елемент, що порівнюється, а цей елемент заноситься у позицію вибраного.

Послідовність кроків:

1. Початкова установка  $j=n$ .

2. Визначення  $\max(k_1, \dots, k_i, \dots, k_j)$ , тобто елемента  $k_i, i = \overline{1, j}$ , який має максимальне значення.
3. Обмін  $k_i \leftrightarrow k_j$ .
4. Зміна  $j$  за формулою  $j=j-1$ .
5. Порівняння. Якщо  $j > 1$  – перейти до п. 2, у протилежному випадку – до п. 6.
6. Кінець сортування.

### *Сортування методом бульбашки*

При цьому методі перший елемент порівнюється по черзі з рештою елементів. З масиву вибирається елемент, значення якого менше від першого. Відбувається обмін елементами. Процедура повторюється доти, поки на першому місці не буде найменший елемент. Для другого елемента перегляд повторюється за тим же принципом. Процедура сортування завершується після  $n-1$ -го перегляду.

Послідовність кроків:

1. Початкова установка номера елемента, який порівнюється з іншими,  $j=1$ .
2. Визначення номера елемента, який порівнюється з елементом п. 1,  $i=j+1$ .
3. Порівняння. Якщо  $k_i < k_j$  – перейти до п. 4, у протилежному випадку – до п. 5.
4. Обмін  $k_i \leftrightarrow k_j$ .
5. Зміна номера елемента  $i$ , з яким порівнюється елемент  $k_j$ , за формулою  $i=i+1$ .
6. Порівняння. Якщо  $i \leq n$  – перейти до п. 3, у протилежному випадку – до п. 7.
7. Зміна номера елемента  $j$ , який порівнюється з рештою елементів, за формулою  $j=j+1$ .
8. Порівняння. Якщо  $j \leq n-1$  – перейти до п. 2, у протилежному випадку – до п. 9.
9. Кінець сортування.

## Лабораторна робота №10. Розв'язування системи лінійних рівнянь методом Гаусса

Зміст завдання: створіть проєкт для розв'язування системи лінійних алгебраїчних рівнянь методом Гаусса. Для перевірки правильності програми кожний студент складає свою систему рівнянь з чотирма невідомими, які повинні приймати цілі значення.

Робочому вікну проєкту надайте вигляду, показаного на рис. 15.

СИСТЕМА РІВНЯНЬ:

2	x1 +	-3	x2 +	1	x3 +	-1	x4 =	-5
1	x1 +	2	x2 +	-1	x3 +	3	x4 =	14
3	x1 +	0	x2 +	2	x3 +	-1	x4 =	5
1	x1 +	1	x2 +	1	x3 +	1	x4 =	10

Розв'язок:

x1	1
x2	2
x3	3
x4	4

Buttons: Розрахувати, Очистити, ВИХІД

Рис. 15. Вікно програми розв'язування системи рівнянь

Схема алгоритму пошуку розв'язку системи лінійних рівнянь методом Гаусса зображена на рис. 16.

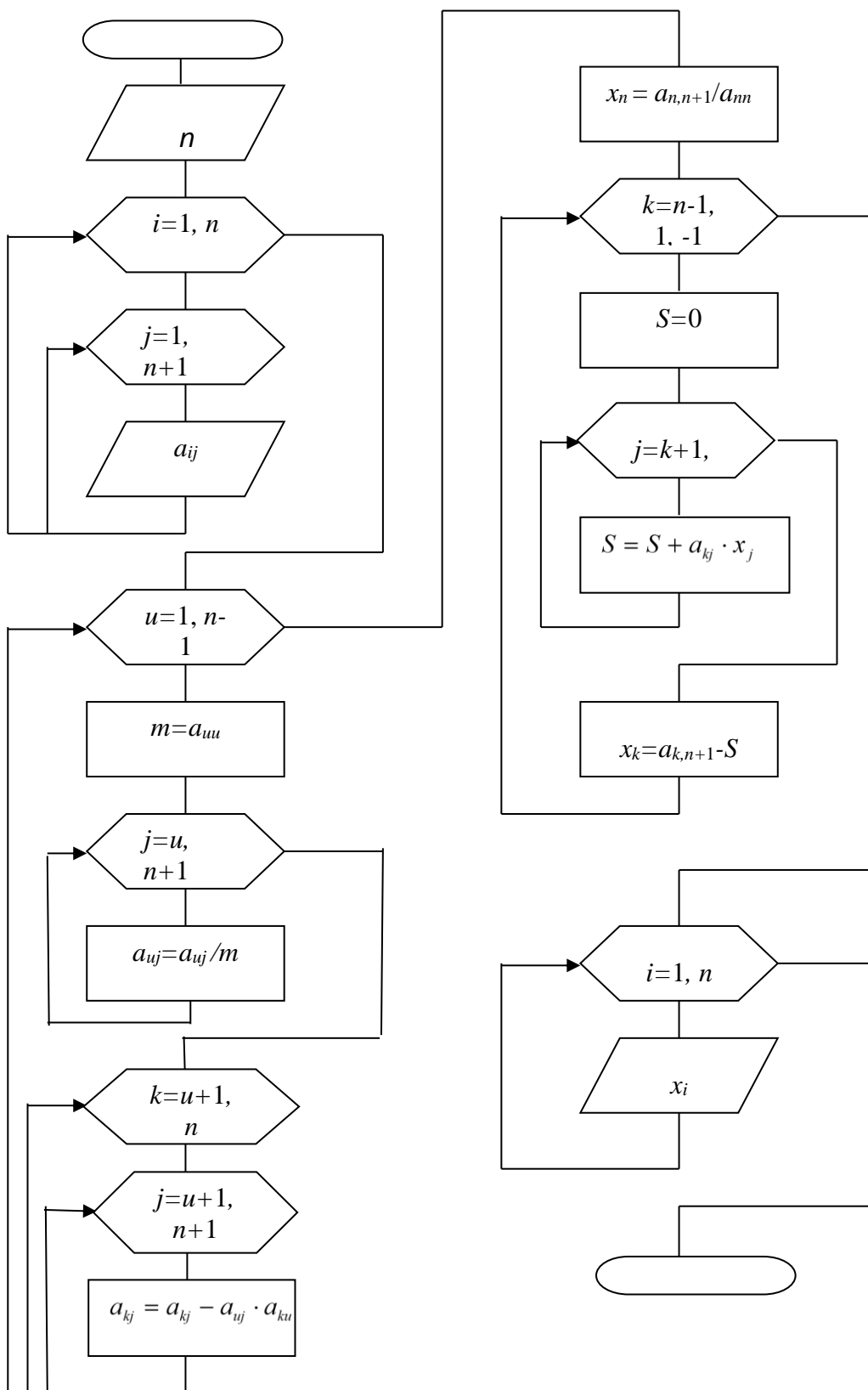


Рис. 16. Схема алгоритму методу Гаусса

## СПИСОК ЛІТЕРАТУРИ

1. Баженов В.А., Венгерський П.С., Горлач В.М., Шевченко О.М. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник для студ. вищ. навч. закл. – К.: Каравела, 2004. – 462 с.
2. Гетц К., Гилберт М. Программирование на Visual Basic 6 и VBA. Руководство разработчика. – К.: Издательская группа BVH, 2003. – 912 с.
3. Глинський Я.М., Анохін В.Є., Ряжська В.А. Бейсик. Від QBasic до Visual Basic.NET: Навч. посіб. 5-те вид., доп.– Львів: СПД Глинський, 2006. – 192 с.
4. Дибкова Л.М. Інформатика та комп'ютерна техніка: Навч. посібник / Л.М. Дибкова. – 3-є вид., доп. – Київ: Академвидав, 2012. – 464 с.
5. Дудзяний І.М. Програмування мовою Visual Basic .NET. Навчальний посібник. –Львів: Видавничий центр ЛНУ імені Івана Франка, 2006. – 272 с.
6. Інформатика: Комп'ютерна техніка. Комп'ютерні технології: Посібник / В.В. Браткевич та ін.; За ред. О.І. Пушкаря. – К.: Видавничий центр “Академія”, 2003. – 703 с.
7. Малачівський П.С. Програмування в середовищі Visual Basic: Навч. посібник. – Львів: Вид-во «Бескид Біт», 2004. – 260 с.

## ДЛЯ ПОДАТОК

## ДЛЯ ПОДАТОК



Навчально-методичне видання

# ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ

Методичні вказівки  
до виконання лабораторних робіт  
для студентів спеціальності  
123 «Комп'ютерна інженерія»

Укладач **КЛЮЄВА** Вікторія Василівна

Комп'ютерне верстання *В.В. Ключової*

Підписано до друку 11.11.2022 Формат 60 × 84<sub>1/16</sub>

Ум. друк. арк. 2,79. Обл.-вид. арк. 1,09.

Електронний документ. Вид № 59/III-17.

Видавець і виготовлювач

Київський національний університет будівництва і архітектури

Повітрофлотський проспект, 31, Київ, Україна, 03680

Свідоцтво про внесення до Державного реєстру суб'єктів  
видавничої справи ДК № 808 від 13.02.2002 р.