

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»  
АВТОМОБІЛЬНО-ДОРОЖНІЙ ІНСТИТУТ

«ЗАТВЕРДЖУЮ»  
Директор АДІ ДВНЗ «ДонНТУ»  
М. М. Чальцев  
8.10.2013 р.

Кафедра «Прикладна математика та інформатика»

**МЕТОДИЧНІ ВКАЗІВКИ  
ДО ПРАКТИКУМУ З РОЗДІЛУ «ОСНОВИ СИСТЕМ  
АВТОМАТИЗАЦІЇ ОБРОБКИ ЕКОНОМІЧНОЇ ІНФОРМАЦІЇ»  
З ДИСЦИПЛІНИ «ЕКОНОМІЧНА ІНФОРМАТИКА» (ДЛЯ  
СТУДЕНТІВ НАПРЯМУ ПІДГОТОВКИ 6.030601  
«МЕНЕДЖМЕНТ» УСІХ ФОРМ НАВЧАННЯ)**

**22/22-2013-02**

«РЕКОМЕНДОВАНО»  
Навчально-методична комісія  
факультету  
«Економіка та управління»  
Протокол № 2  
від 16.10.2013 р.

«РЕКОМЕНДОВАНО»  
Кафедра  
«Прикладна математика та  
інформатика»  
Протокол № 12  
від 14.05.13 р.

УДК 681.3.06(071)

Методичні вказівки до практикуму з розділу «Основи систем автоматизації обробки економічної інформації» з дисципліни «Економічна інформатика» (для студентів напряму підготовки 6.030601 «Менеджмент» усіх форм навчання) [Електронний ресурс] / укладачі: М. Є. Корольов, Р. С. Кравченко. – Електрон. дані. – Горлівка: ДВНЗ «ДонНТУ» АДІ, 2013. – 1 електрон. опт. диск (CD-R); 12 см. – Систем. вимоги: Pentium; 32 MB RAM; WINDOWS XP/Vista/7; MS Word 2000–2010. – Назва з титул. екрану.

Методичні вказівки мають на меті дати студенту загальну уяву щодо сучасних економічних інформаційних систем, тенденцій їх розвитку й конкретних реалізацій; сформувати навички роботи з практичними інструментами спеціаліста – програмними комплексами та інформаційними ресурсами, що необхідні при обробці економічної та управлінської інформації.

Методичні вказівки є частиною мультимедійного навчального комплексу «АРМ викладач-студент», який містить додатковий матеріал, що необхідний для виконання практикуму.

Укладачі: Корольов М. Є., к.ф.-м.н., доц.  
Кравченко Р. С.

Відповідальний за випуск: Хребет В. Г., к.ф.-м.н., доц.

Рецензент: Вовк Л. П., д.т.н., проф.

© Державний вищий навчальний заклад  
«Донецький національний технічний університет»  
Автомобільно-дорожній інститут, 2013

## ЗМІСТ

ВСТУП .....	4
1 СТВОРЕННЯ ЗАВАНТАЖУВАЛЬНОЇ ФОРМИ ПРОЕКТУ .....	5
2 МАСИВИ .....	13
2.1 Оголошення масиву .....	13
2.2 Розмірність масиву, робота з вимірами .....	14
2.3 Створення масиву .....	16
2.4 Відображення елементів масиву з використанням компонента DataGridView .....	17
2.5 Реалізація моделі «Вектор» .....	18
2.6 Реалізація моделі «Матриця-1» .....	26
2.7 Діалогові вікна роботи із зовнішніми файлами .....	33
2.8 Реалізація моделі «Матриця-2» .....	38
3 ПРОЦЕДУРИ У VB .....	47
3.1 Виклик процедури та повернення з неї .....	47
3.2 Параметри й аргументи. Типи процедур .....	47
3.3 Процедури й структурований код .....	48
3.4 Створення процедури .....	48
3.5 Реалізація моделі «Процедура – Sub» .....	51
3.6 Реалізація моделі «Процедура – Function» .....	58
4 ГРАФІЧНІ МОЖЛИВОСТІ МОВИ VB .....	63
4.1 Загальні відомості про векторну графіку .....	63
4.2 Реалізація моделі «Графіка» .....	68
ДОДАТОК А ДОДАТКОВІ ВІДОМОСТІ З VB .....	85
А.1 Блок-схеми алгоритмів .....	85
А.2 Терміни й поняття .....	87
ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ .....	102

## ВСТУП

Основною метою викладання дисципліни «Економічна інформатика» є формування у студентів знань щодо принципів обробки економічної інформації, а також проектування систем автоматизації цього процесу. Навчальна дисципліна розрахована на студентів усіх форм навчання, які засвоїли курс з інформатики й мають навички роботи на персональному комп'ютері, володіють знаннями щодо роботи з прикладними пакетами офісних програм, обізнані в основах алгоритмізації процесів обробки первинної інформації.

Мета практикуму з дисципліни «Економічна інформатика» полягає у вивченні теоретичних основ економічної інформатики, набутті навичок використання прикладних систем оброблення економічних даних та створення систем, що автоматизують цей процес, який покладено в основу дослідження соціально-економічних систем та розв'язання завдань фахового спрямування; у набутті вмінь і навичок зі створення додатків для роботи з масивами, як сукупності економічних даних, із зовнішніми файлами, у яких можуть зберігатися ці дані, та при візуалізації різноманітних економічних процесів.

Предмет: засоби автоматизації інформаційних процесів із використанням економічних даних.

Кінцевим результатом виконання практикуму є комплекс автоматизації обробки економічної інформації.

Методичні вказівки є частиною мультимедійного навчального комплексу «АРМ викладач-студент», що представляє собою додаток, який містить компоненти, які необхідні для виконання практичних робіт із дисципліни «Економічна інформатика»:

– демонстраційний програмний комплекс, усі етапи створення та тестування якого, викладені в даних методичних вказівках;

– мультимедійний курс із відеоуроками, що охоплюють найбільш важливі етапи створення системи автоматизації обробки економічної інформації;

– комплекс тестових завдань для контролю знань студентів у вигляді мультимедійного додатка;

– задачі для індивідуального виконання та програмний генератор листа завдань.

# 1 СТВОРЕННЯ ЗАВАНТАЖУВАЛЬНОЇ ФОРМИ ПРОЕКТУ

## Елемент керування ToolStrip

Для проектування панелей інструментів використовується елемент керування ToolStrip. Як правило, у сучасних Windows-додатках, панель інструментів розташовується вгорі вікна під головним меню, хоча її можна переносити та змінювати розміри. Звичайно панель інструментів містить команди головного або контекстного меню, які використовуються найчастіше.

Панель інструментів складається з набору елементів керування ToolStripButton, які можуть мати вигляд спадаючого меню, кнопок із зображенням або текстом і роздільників. Для додавання елементів на панель інструментів використовується діалогове вікно «Редактор колекції елементів» (рисунок 1.1), що відкривається при виборі властивості Items (пункти).

У нашій роботі ми плануємо реалізувати наступні моделі

- ✚ Вектор;
- ✚ Матриця1;
- ✚ Матриця2;
- ✚ Процедура;
- ✚ Процедура-функція;
- ✚ Графіка.

Таким чином, використовуючи кнопку «Додати», додаємо на панель керування ToolStrip1, члени ToolStripButton1,...,ToolStripButton6, які відповідні до вищеперелічених моделей.

Використовуючи ToolStripSeparator (separator-роздільник), створимо візуальний образ угруповання кнопок за призначенням, а саме: робота з масивами, підпрограмами, графікою.

За кожним членом ToolStripButton закріпимо виклик відповідної форми, яку додамо з використанням команди: «Додати форму Windows» (рисунок 1.2.) Для вибору зображення на ToolStripButton використовуйте властивість Image (зображення).

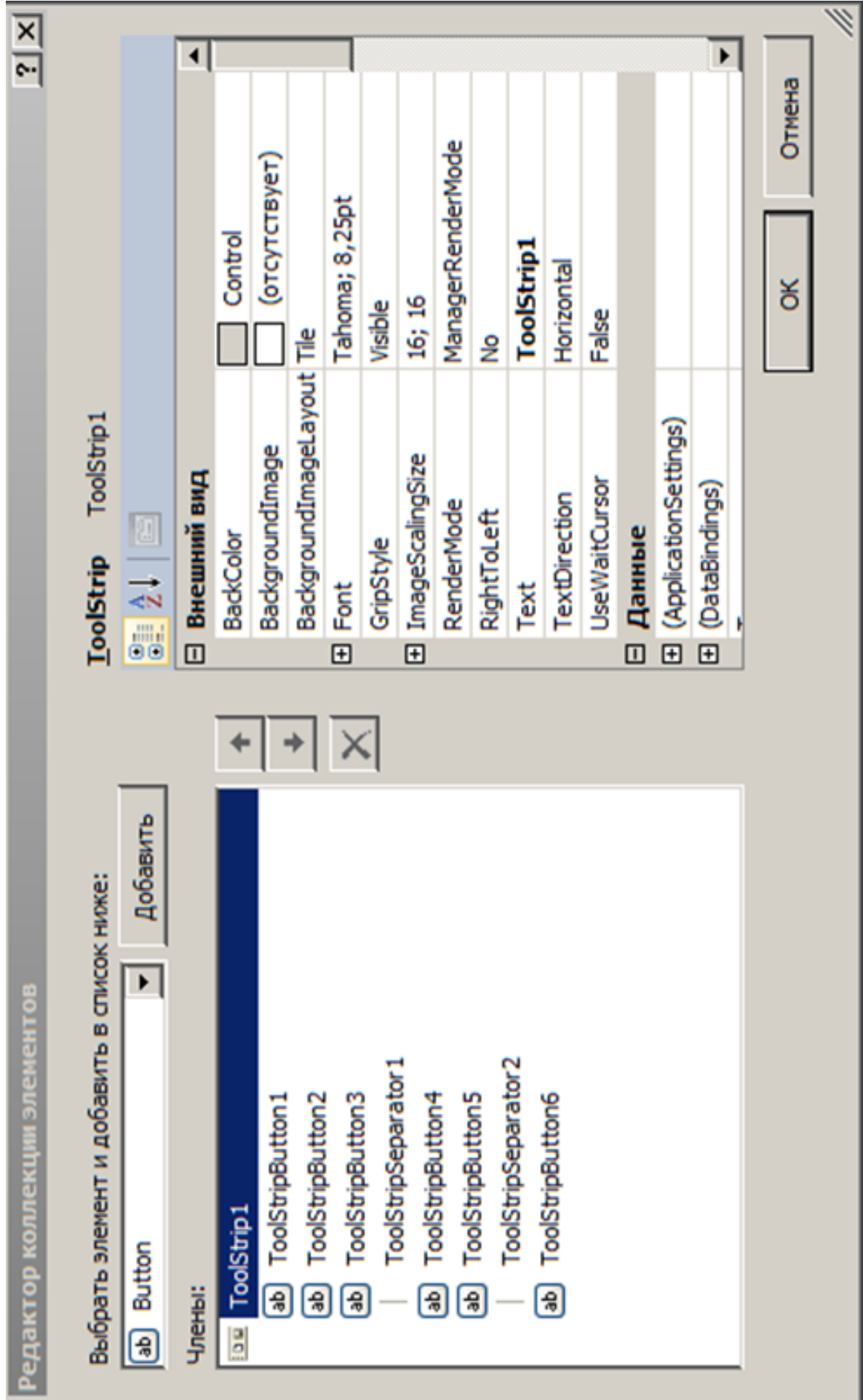


Рисунок 1.1 – Диалогове вікно редактора колекції елементів

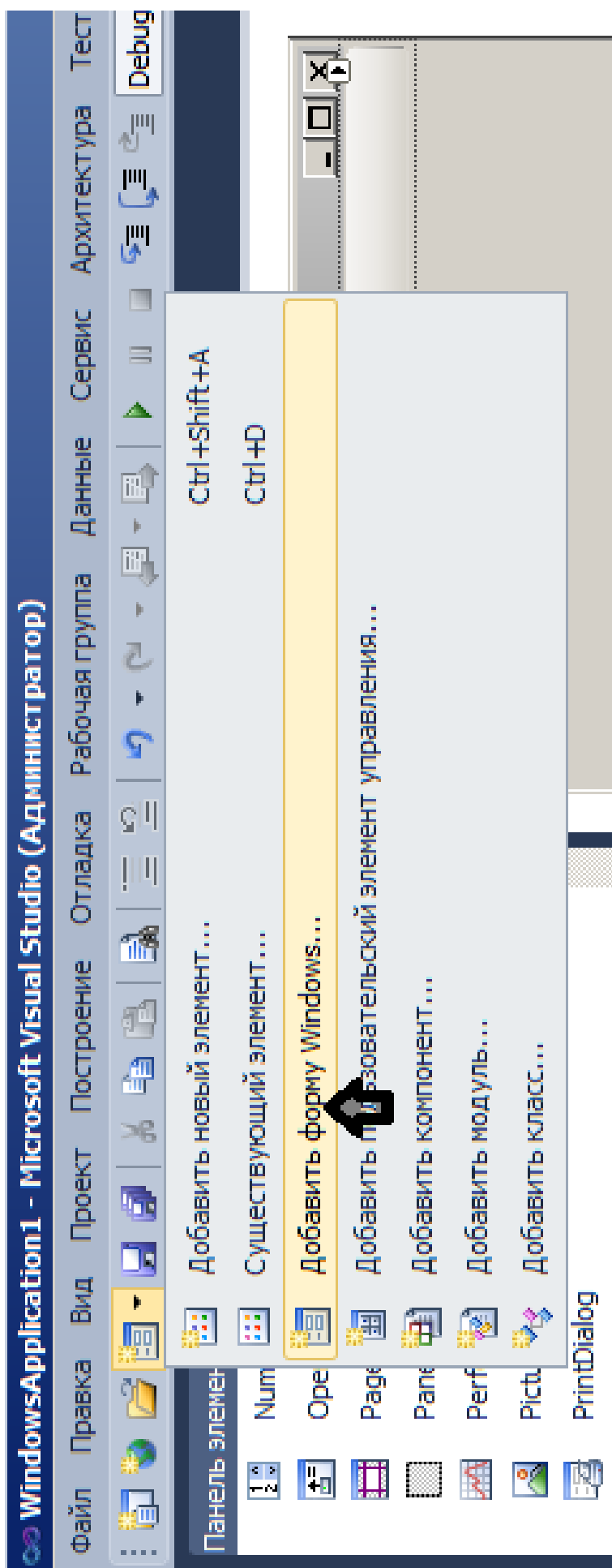


Рисунок 1.2 – Добавления в проект новой Windows Form

Алгоритм запуску кожної моделі проекту з головної (Main) завантажувальної форми, реалізований на наступній схемі:

Лістинг програми до рисунка 1.3 має такий вигляд:

---

```
Public Class Main
```

```
Private Sub ToolStripButton6_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles ToolStripButton6.Click  
    Графика.Show()  
End Sub
```

```
Private Sub ToolStripButton3_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles ToolStripButton3.Click  
    Матрица_2.Show()  
End Sub
```

```
Private Sub ToolStripButton1_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles ToolStripButton1.Click  
    Вектор.Show()  
End Sub
```

```
Private Sub ToolStripButton2_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles ToolStripButton2.Click  
    Матрица_1.Show()  
End Sub
```

```
Private Sub ToolStripButton4_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles ToolStripButton4.Click  
    Подпрограмма_SUB.Show()  
End Sub
```

```
Private Sub ToolStripButton5_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles ToolStripButton5.Click  
    Подпрограмма_Function.Show()  
End Sub
```

```
End Class
```

---



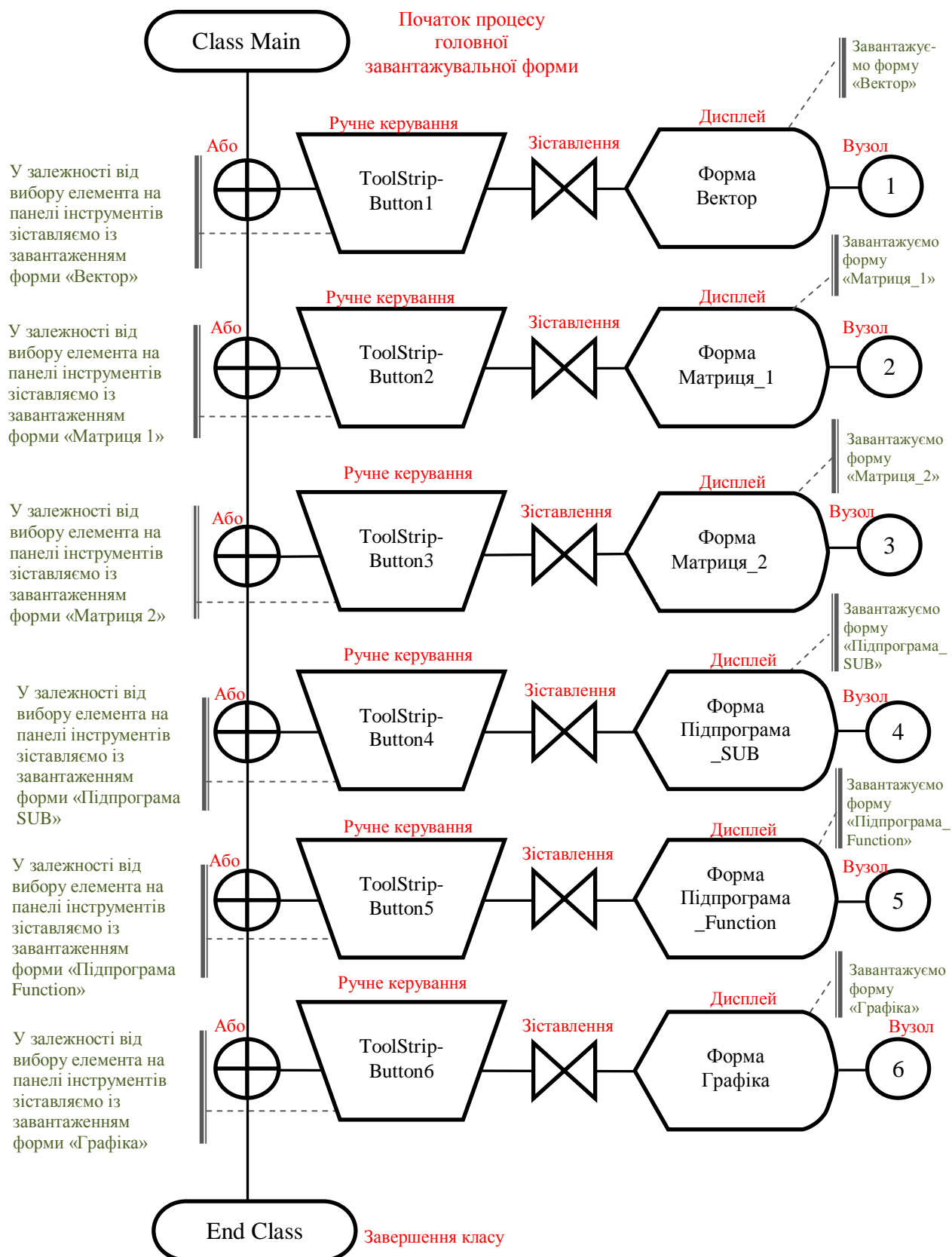


Рисунок 1.3 – Блок-схема головної (Main) завантажувальної форми

Таким чином, ми створили «Главную загрузочную форму» (рисунок 1.4), при цьому «Обозреватель решений» буде мати вигляд, що показаний на рисунку 1.5.

Зауваження: доцільно на головну завантажувальну форму для зручності використання помістити `LinkLabel` (відображає гіперпосилання), за яким закріпити файл опису роботи: «Описание работы.doc».

Програмний код даної ділянки програми має наступний вигляд:

---

```
Dim sss As String = My.Application.Info.DirectoryPath  
System.Diagnostics.Process.Start(sss & "\Пояснительная записка.doc")
```

---

У розглянутому прикладі, `sss` повертає каталог, в якому перебуває додаток, `System.Diagnostics.Process.Start` запускає ресурс процесу шляхом вказівки імені документа або файла.

Зауваження: метод `System.Diagnostics.Process.Start` запускає ресурс процесу та зв'язує його з компонентом `Process`.

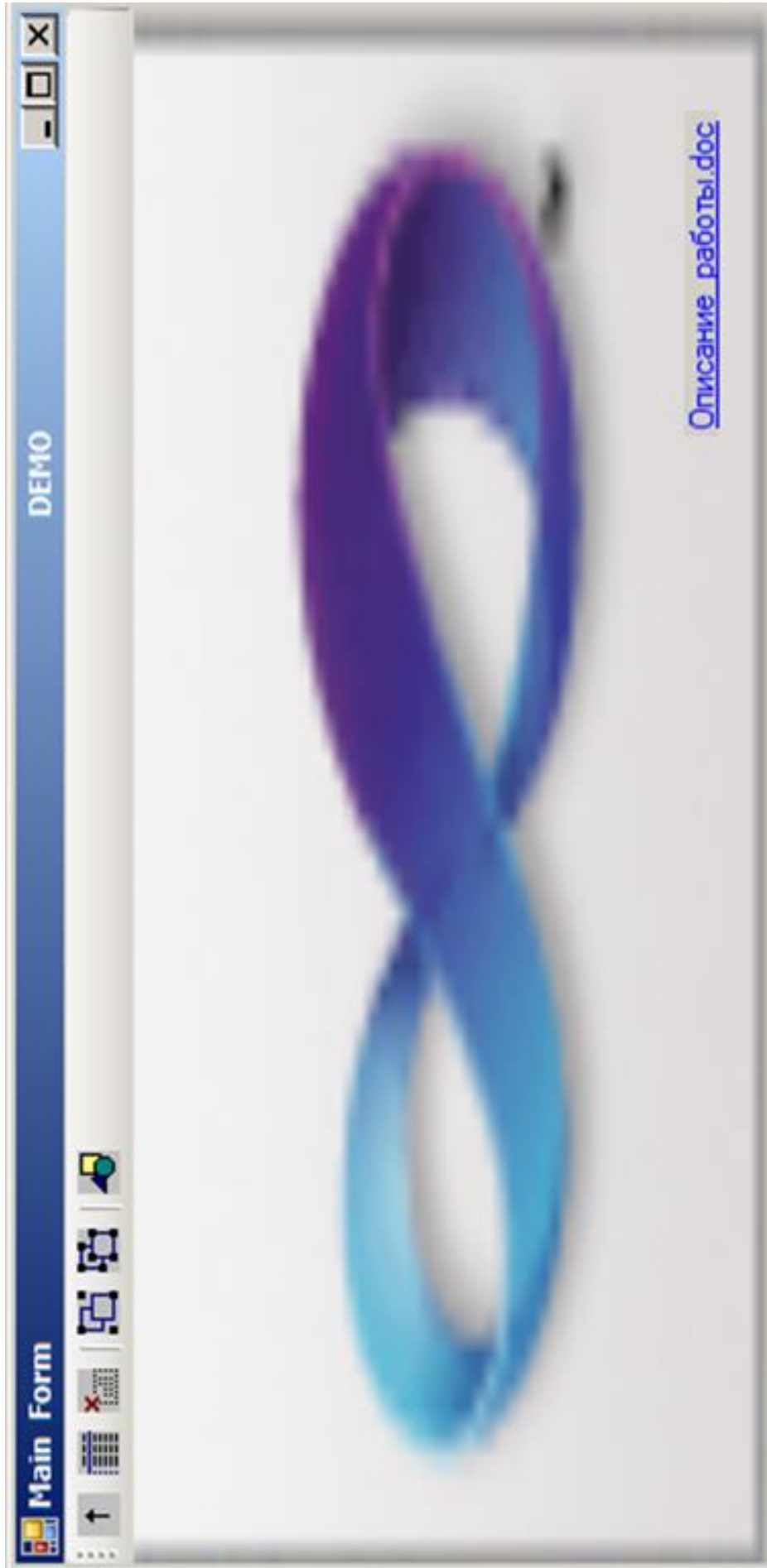


Рисунок 1.4 – Запуск головной формы Main Form

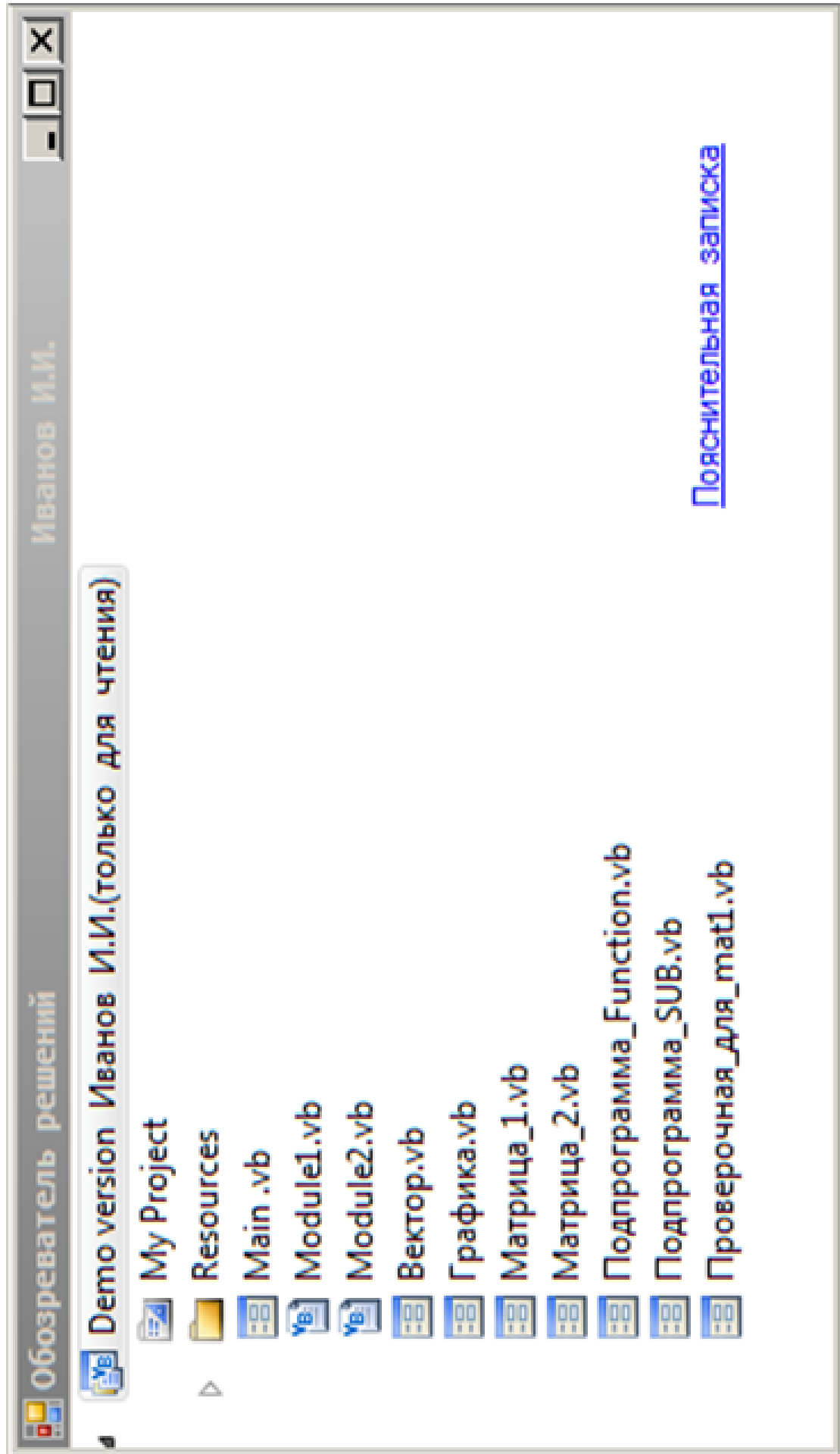


Рисунок 1.5 – Зовнішній вигляд шаблону «Обозревателя решений»

## 2 МАСИВИ

Масив представляє собою набір значень, що логічно зв'язані один з одним, наприклад, число студентів у кожній групі.

Масив дозволяє посилатися на ці зв'язані значення по одному імені й використовувати номер, що називається індексом або підіндексом, щоб відрізнити елементи один від іншого. Окремі значення називаються елементами масиву. Вони представляються безперервною послідовністю від індексу 0 до найбільшого значення індексу.

На відміну від масиву, змінна, що містить одиночне значення, називається скалярною змінною.

### 2.1 Оголошення масиву

Змінна масиву оголошується так само, як будь-яка інша змінна за допомогою інструкції Dim. За іменем змінної додаються дужки, що показує, що це масив, а не скаляр (змінна, що містить одне значення). Для оголошення змінної одновимірного масиву необхідно додати одну пару дужок після імені змінної.

У наступному прикладі оголошується змінна масиву для зберігання числа учнів у кожній групі:

**Dim Students (5) As Integer**

**Dim As Integer** Масив «Students» у попередньому прикладі містить 6 елементів. Індеси елементів знаходяться в діапазоні від 0 до 5. Використовувати масив простіше, ніж оголосити 6 окремих змінних.

На наступному рисунку надано структуру масиву «Students»:



Рисунок 2.1 – Елементи масиву «Students»

У наступному прикладі показано звертання до першого (нульового), другого й останнього елементу введеного масиву «Students» (рисунок 2.2).

---

```

Dim students(5) As Integer
For i = 1 To 5
    students(i) = InputBox("Введите количество студентов на " & Chr(13) & i & "-ом курсе")
Next
Dim студ_подг_отдел As Integer = students(0)
Dim студ_первого_курса As Integer = students(1)
Dim студ_пятого_курса As Integer = students(5)
MsgBox("Студентов в подготовительном отделении: " & CStr(студ_подг_отдел))
MsgBox("Студентов на первом курсе:" & CStr(студ_первого_курса))
MsgBox("Студентов на пятом курсе: " & CStr(студ_пятого_курса))

```

---

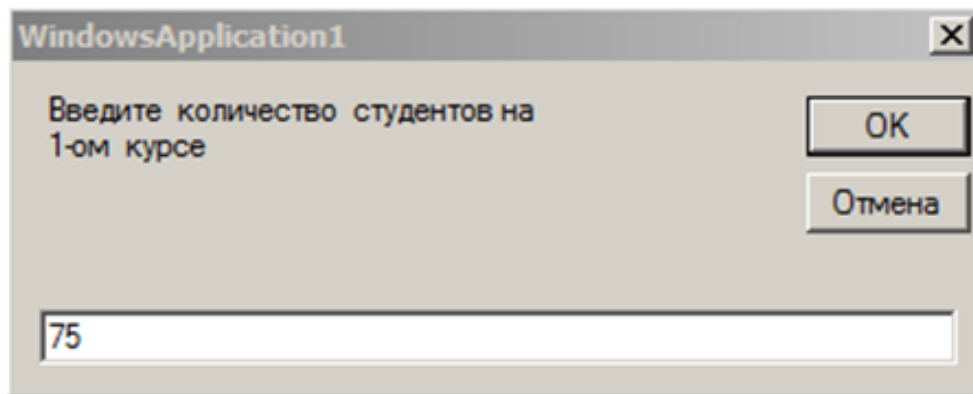


Рисунок 2.2 – Діалогове вікно введення елементів масиву

## 2.2 Розмірність масиву, робота з вимірами

Розмірність – це напрямок, в якому можна змінювати специфікацію елементів масиву. Масив «Students» у попередньому прикладі використовує один індекс і вважається одномірним (рисунок 2.3). Масив, що використовує більше одного індексу, називається багатомірним масивом.

Рангом масиву називається число його вимірів. Необхідно вказувати елемент масиву, указавши індекс для кожного з його вимірів. Елементи безперервні вздовж кожного виміру, починаючи з індексу 0 до найбільшого індексу для цього виміру.

На рисунках 2.4–2.5 показано загальну структуру масивів різного рангу. Кожний елемент на ілюстраціях указує значення індексу при звертаннях до нього. Наприклад, можна одержати доступ до першого елемента другого рядка двовимірного масиву, указавши індекси (1, 0).

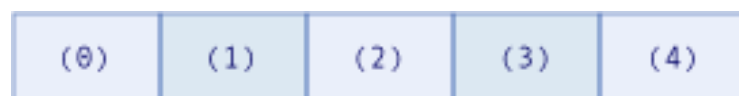


Рисунок 2.3 – Одномірний масив

(0,0)				(0,4)
(1,0)				
(3,0)				(3,4)

Рисунок 2.4 – Двовимірний масив

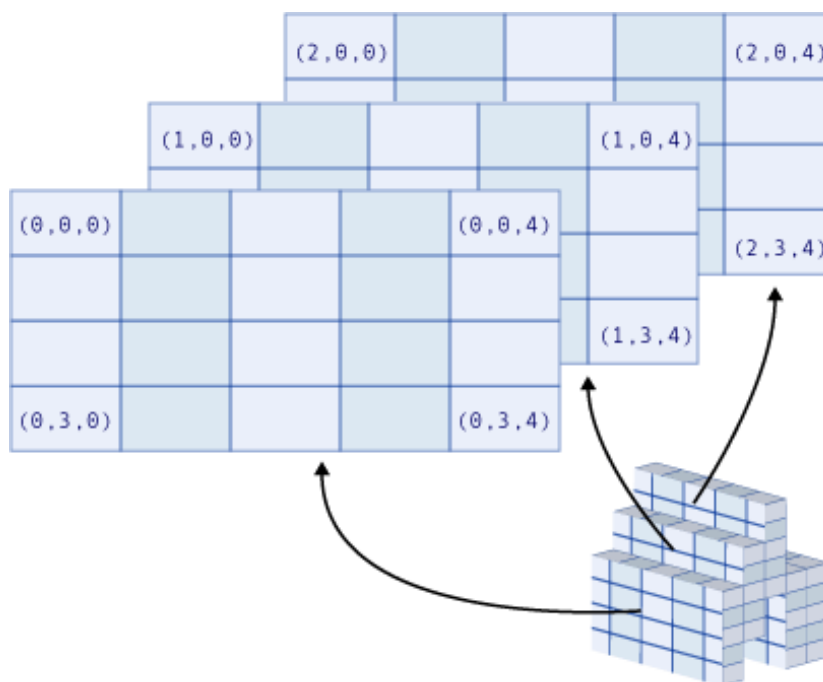


Рисунок 2.5 – Тривимірний масив

### Два виміри

Деякі масиви мають два виміри. Наприклад, отримані оцінки в плинні семестру для кожного студента. Специфікація елемента вимагає одночасно вказівки номера студента в списку групи й саме отриманої оцінки по даній роботі. Кожний елемент містить лічильник для цієї комбінації. Таким чином, такий масив використовує два індекси.

```
Dim officecounts(40, 5) As Byte
```

Двовимірний масив також називають прямокутним масивом.

### Три виміри

Деякі масиви мають три виміри. Наприклад, координати в тривимірному просторі. Такий масив використовує три індекси, які в цьому випадку представляють координати фізичного простору  $x$ ,  $y$  і  $z$ . У наступному прикладі оголошується змінна для зберігання тривимірного масиву координат вектора в тривимірному просторі:

```
Dim вектор(100, 100, 100) As Double
```

## 2.3 Створення масиву

Масив можна створити двома способами. Можна вказати розмір масиву після його оголошення або, оскільки масив є об'єктом, можна створити його з оператором «New» і привласнити його змінній масиву. Можна виконати це в рамках оголошення масиву або в наступній інструкції призначення, як показано в наступних прикладах.

Приклад 1. `Dim a(100) As Integer`

Приклад 2. `Dim a() As Integer = {1, 2, 3}`

Приклад 3. `Dim x() = {1, 2.2, "3"}`

Приклад 4. Можна створити багатомірний масив за допомогою вкладених літералів масиву.

`Dim Матрица = {{1, 2}, {3, 4}}`

Приклад 5. Для оголошення змінної багатомірного масиву необхідно додати одну пару дужок, усередині дужок розташувати коми, що розділяють виміри масиву.

---

```
Dim arrA = New Double(,) {{1, 2, 3, 4, 5}, {6, 7, 8, 9, 10}, _
    {11, 22, 33, 44, 55},{4, 5, 6, 5, 6}, {0, 0, 0, 0, 0}}
```

```
For i = 0 To 4
    For j = 0 To 4
        TextBox1.Text = TextBox1.Text & vbTab & arrA(i, j)
    Next
    TextBox1.Text = TextBox1.Text & vbNewLine
Next
```

---

При запуску програми одержимо наступну матрицю (рисунок 2.6):

1	2	3	4	5
6	7	8	9	10
11	22	33	44	55
4	5	6	5	6
0	0	0	0	0

Рисунок 2.6 – Результат роботи програми

Приклад 6.

Оператор **ReDim** перерозподіляє область пам'яті для зберігання змінної масиву.

Компонент **Preserve** використовується для зберігання даних існуючого масиву при зміні розміру тільки останньої розмірності масиву.

Наведемо демонстраційний приклад (рисунок 2.7):



```

Dim матрица(2, 3) 'идентифицируем двумерный массив
For i = 1 To 2
    For j = 1 To 3
        'вводим поэлементно массив - матрица(2, 3)
        матрица(i, j) = InputBox("Введите поэлементно массив данных", , 0)
        'введенные данные выводим в многострочное текстовое поле
        TextBox1.Text = TextBox1.Text & vbTab & матрица(i, j)
    Next
    TextBox1.Text = TextBox1.Text & vbNewLine
Next
'
ReDim Preserve матрица(2, 4) 'создаётся новый массив с новой _
    размерностью в который при помощи ключевого слова Preserve _
    копируются старые данные введенные выше
матрица(2, 4) = 999
For i = 1 To 2
    For j = 1 To 4 'столбцов на единицу больше
        TextBox2.Text = TextBox2.Text & vbTab & матрица(i, j)
    Next
    TextBox2.Text = TextBox2.Text & vbNewLine
Next

```

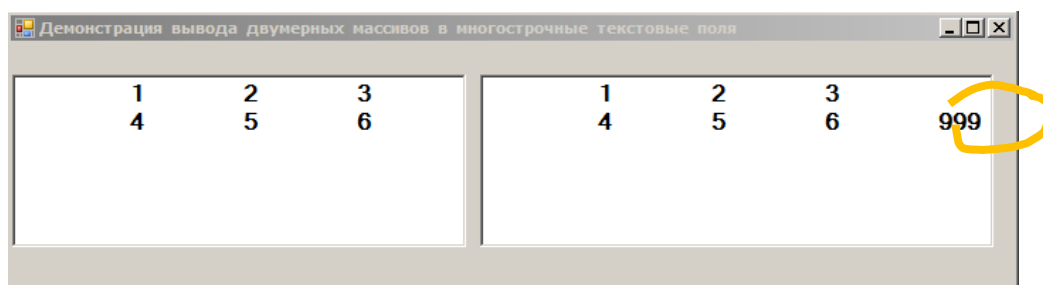



Рисунок 2.7 – Демонстраційний приклад

## 2.4 Відображення елементів масиву з використанням компонента DataGridView

Елемент керування DataGridView  для відображення даних створює таблицю, що настроюється. Клас DataGridView допускає налаштування комірок, рядків, стовпців і границь.

У моделі «Вектор» нашої роботи ми будемо використовувати наступні властивості:

1. DataGridView.RowCount повертає або задає число рядків, що відображуються в об'єкті DataGridView.

2. DataGridView.ColumnCount повертає або задає число стовпців, що відображуються в об'єкті DataGridView.

3. DataGridView.Item надає індикатор для одержання або завдання комірки, що розташована на перетинанні зазначених рядка й стовпця.

4. `DataGridView.AutoSizeColumnsMode` повертає або задає значення, що вказує, як визначається ширина стовпця.

Наведемо приклад із використанням даних властивостей (рисунок 2.8):

---

```
With DataGridView1
    .ColumnCount = 5 + 1
    .RowCount = 3
    .AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.DisplayedCells
    'где .DisplayedCells - ширина столбцов изменяется так, чтобы вместить _
    'содержимое всех ячеек столбцов отображающихся на экране в настоящий момент.
    .Item(4, 2).Value = "Я ячейка (4,2)" '4-индекс столбца, 2- индекс строки.
    .Item(0, 0).Value = "Я ячейка (0,0)" ' ВНИМАНИЕ: индексация начинается с 0.
    .Columns.Item(1).HeaderText = "Я 1-й столбец" ' где .HeaderText -задает _
    ' текст ячейки заголовка столбца.
    .Columns.Item(0).HeaderText = "Я 0-й столбец"
End With
```

---

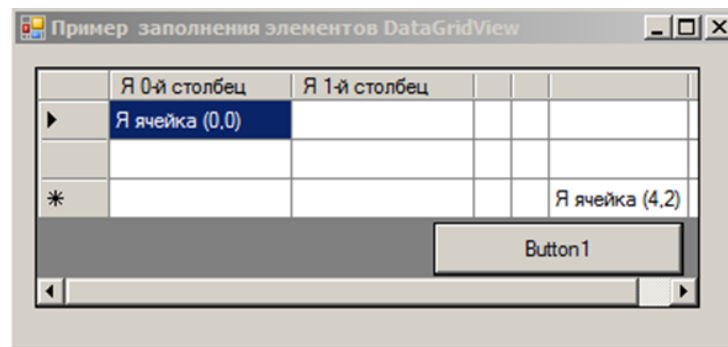


Рисунок 2.8 – Форма з елементом керування DataGridView

Зауваження: у деяких випадках на формі потрібно відобразити тільки таблицю (матрицю) або рядок матриці (вектор). Для цього можна використовувати елемент `DataGridView`, у якому сховані нульовий стовпець (`.Columns.Item(0).Visible = False`, дивіться модель «Вектор») або нульовий рядок (`.Rows(0).Visible = False`). Також можна не відображати заголовки стовпців (`.ColumnHeadersVisible = False`) і заголовки рядків (`.RowHeadersVisible = False`), дивіться модель «Матриця-2».

## 2.5 Реалізація моделі «Вектор»

Умова. Аналітичний відділ мережі автосервісів «Омега-моторс» розробив програму технічної модернізації комплексів автомийок та склав договір із компанією-постачальником відповідного обладнання `SONNY'S Enterprises Inc.` За умовою договору «Омега-моторс» отримала гібридну тунельну систему `SONNY'S 60-MT` для тестування в реальних умовах строком на 3 місяці. Протягом періоду тестової експлуатації системи були зібрані наступні дані про темпи зростання витрат (тис. у. о.) на надання послуг із мийки авто:

$$a_1 = \ln \frac{1}{2}, a_2 = \ln \frac{1}{4}, a_3 = \ln \frac{1}{6} \dots$$

де  $a_i$  – значення темпу зростання витрат в  $i$ -й місяць тестової експлуатації (тис. у. о.).

На підставі цих даних, за умовою того, що й у подальшому динаміка показників зростання залишиться незмінною, виявити аналітичну залежність темпів зростання витрат від місяця експлуатації та визначити можливість окупності саме цього обладнання за 15 місяців. У разі позитивного результату зробити висновок про доцільність придбання системи, якщо вартість кожного комплексу SONNY'S 60-MT складає 35 тис. у. о.

Для розв'язання задачі розробити програмний модуль, який дозволяє виконати необхідні обчислення, та представити результати розрахунків у вигляді, що показаний на рисунках 2.9–2.10.

Алгоритм розв'язку (рисунок 2.11) полягає в наступному.

1. Послідовність значень розглянути як члени числового ряду, які формуються згідно з деякою аналітичною залежністю. У даному випадку:

$$u_i = \ln \frac{1}{i - n}, i - n = 2, 4, \dots$$

2. Алгоритм має ітеративний характер, на кожному кроці якого необхідно обчислювати член ряду, формувати одновимірний масив та накопичувати суму.

3. Якщо накопичена сума перевищує за модулем значення 35 тис. у. о., то можна зробити висновок про можливість окупності обладнання у 15-ти місячний термін.

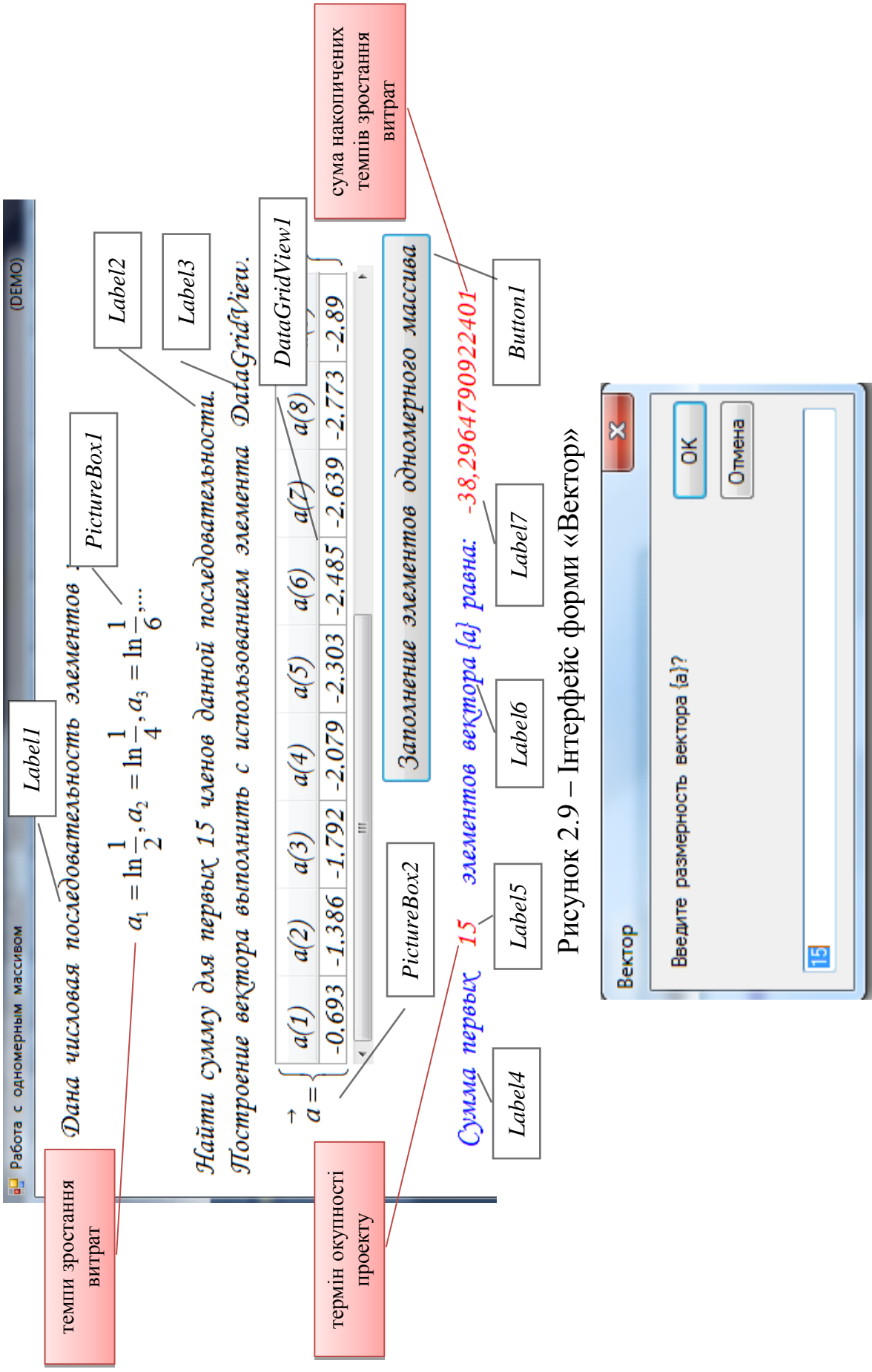


Рисунок 2.9 – Интерфейс формы «Вектор»

Рисунок 2.10 – Диалоговое окно «Вектор»

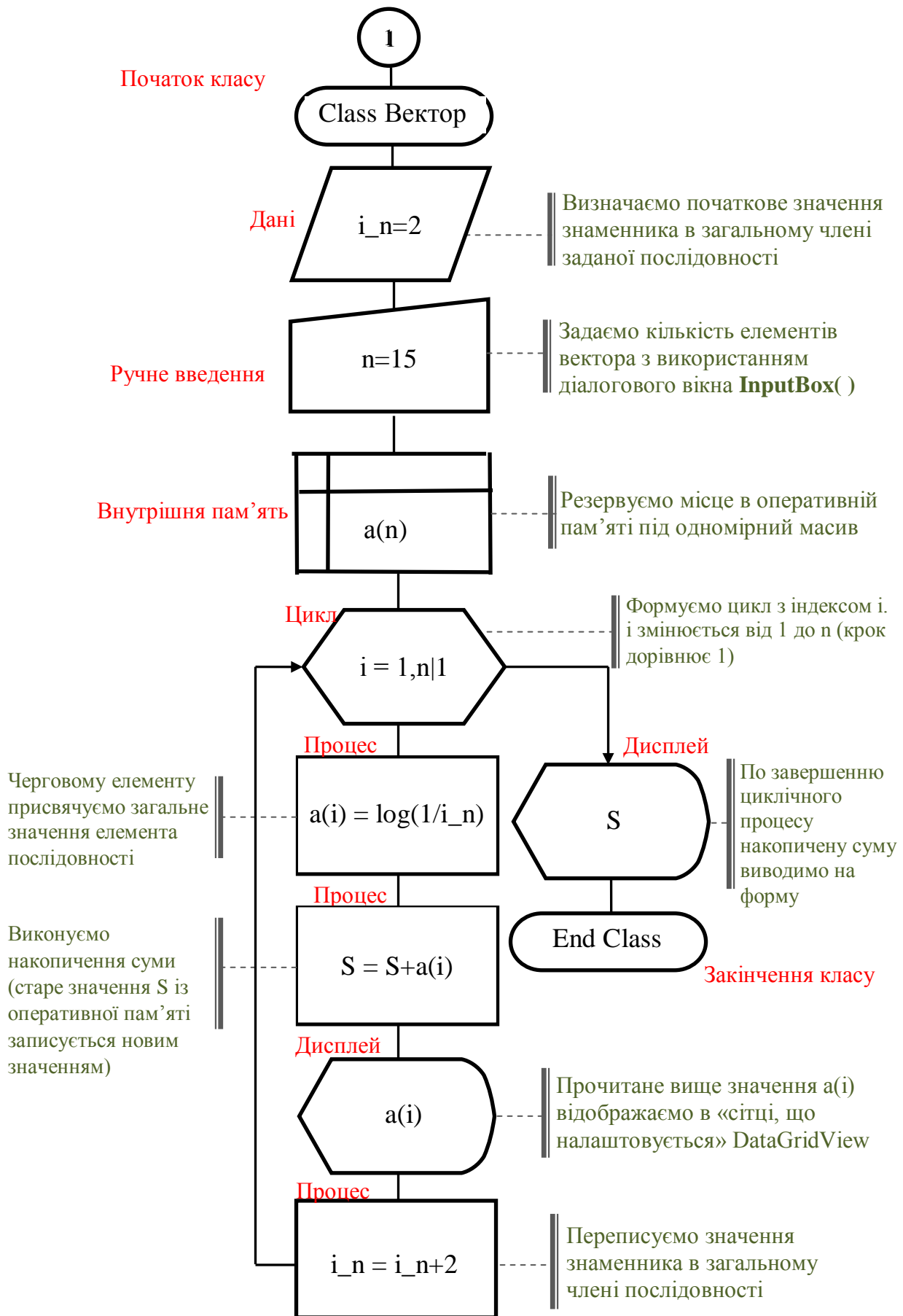


Рисунок 2.11 – Блок-схема програми моделі «Вектор»

## Лістинг програми моделі «Вектор»

---

```
Imports System.Math

Public Class Вектор
    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        Dim i As Integer, n As Integer, i_n As Integer = 2, s As Double
        n = InputBox("Введите размерность вектора {a}?", "Вектор", 15)
        Label5.Text = n
        Dim a(n) As Double
    With DataGridView1 'DataGridView-Отображает данные в настраиваемой сетке.

        .RowHeadersVisible = False 'RowHeadersVisible-Возвращает или задает
            значение, указывающее, отображается ли
            столбец, содержащий заголовки строк.

        .RowCount = 1 'Возвращает или задает число строк, отображае
            мых в объекте DataGridView.
        .AutoSizeColumnsMode = DataGridViewAutoSizeCoumnsMode.AllCells
            'AutoSizeColumnsMode-Возвращает или
            задает значение, указывающее, как
            определяется ширина столбца.
        .Columns.Item(0).Visible = False 'Columns-Получает коллекцию,
            содержащую все столбцы в элементе управления.
            '
            Item-<Ru.Элемент>
        For i = 1 To n 'Реализуем цикл по координатам вектора {a}
            .Columns.Add(i, "a(" & i & ")") 'где i- имя по которому
                будет упоминаться столбец
            '
            "a(" & i & ")"- текст заголовка столбца
            a(i) = Log(1 / i_n) ' Очередному элементу одномерного
                массива присваиваем общий член ряда (см.условие)
            s = s + a(i)
            .Item(i, 0).Value = Round(a(i), 3) 'DataGridView.Item-Предоставляет
                индекатор для получения или задания яки, расположенной
                на пересечении указанных строки и столбца. Value-
                получает или задает значение, связанное с этой ячейкой.
                где: i- индекс столбца содержащего ячейку
                0 - индекс строки содержащей ячейку, (т.е. исходная
                таблица содержит только одну строку)
            i_n = i_n + 2 ' шаг индекса общего члена ряда = 2
        Next
    End With ' конец блока With
    Label7.Text = s
    End Sub
End Class
```

---

У наступному розділі нам знадобляться додаткові можливості форматування DataGridView, а також заповнення двовимірного масиву через елемент керування DataGridView.

Розглянемо докладно ці питання:

## Форматування елемента керування DataGridView

DataGridViewCellStyle подає відомості про форматування й стиль, що застосовуються до окремих комірок в елементі керування DataGridView.

DataGridView.Columns.Clear() очищує колекцію, у цьому випадку стовпці.

DataGridView.Rows.Clear() очищає колекцію, у цьому випадку рядки.

DataGridView.GridColor – властивість, яка повертає або задає колір ліній сітки, що розділяють комірки об'єкта DataGridView.

Color.FromArgb – метод, який створює структуру Color із чотирьох значень компонентів ARGB (alpha, червоний, зелений і синій). Щоб створити непрозорий колір, установіть для alpha значення 255. Щоб створити напівпрозорий колір, установіть для alpha будь-яке значення від 1 до 254.

DataGridView.BackgroundColor повертає або задає колір тла для об'єкта DataGridView.

DataGridView.DefaultCellStyle – властивість, яка повертає або задає стиль комірки за замовчуванням, який буде застосовуватися до комірок в об'єкті DataGridView, якщо не задані які-небудь інші властивості стилю комірки.

### Демонстраційний приклад 1

З використанням елемента DataGridView створити шаблон бланка успішності групи студентів із математики, інформатики та історії. Виконати форматування стовпців і сітки даних (рисунок 2.12).

Лістинг програми демонстраційного прикладу 1 буде мати такий вигляд:

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        Dim format_математика As New DataGridViewCellStyle
        Dim format_информатика As New DataGridViewCellStyle
        Dim format_история As New DataGridViewCellStyle
        format_математика.BackColor = Color.Gold
        format_информатика.BackColor = Color.GreenYellow
        format_история.BackColor = Color.Khaki
        With DataGridView1
            .RowCount = 7
            .Columns.Item(0).HeaderText = "Ф.И.О."
            .Columns.Add(1, "математика")
            .Columns.Add(2, "информатика")
            .Columns.Add(3, "история")
            .Columns(1).DefaultCellStyle = format_математика
            .Columns(2).DefaultCellStyle = format_информатика
            .Columns(3).DefaultCellStyle = format_история
            .GridColor = Color.OrangeRed
        End With
    End Sub
End Class
```

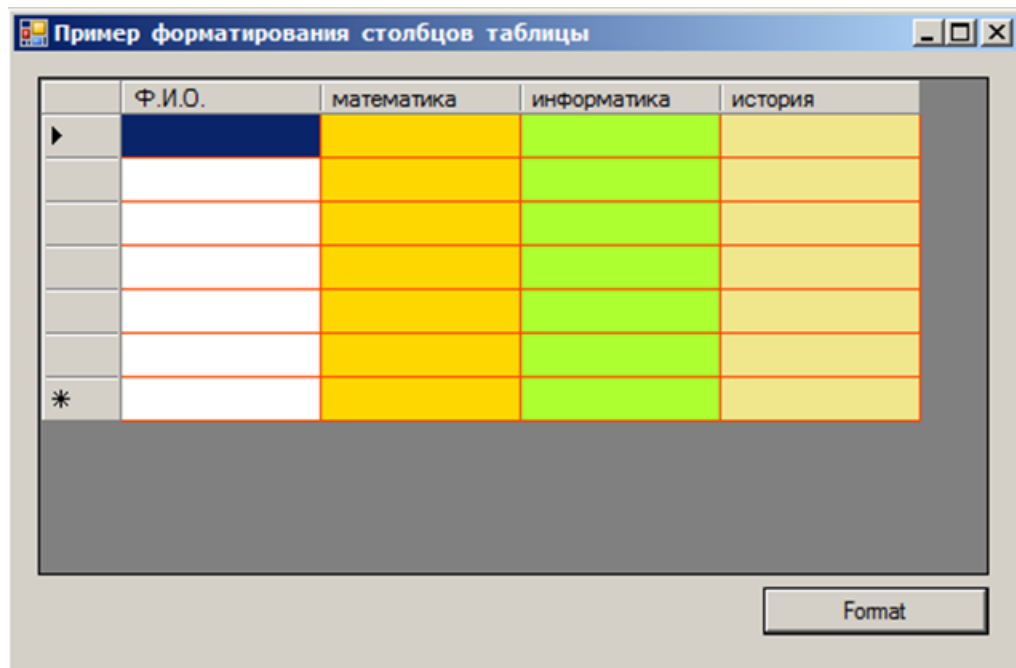


Рисунок 2.12 – Результат роботи програми

Зчитування значень сітки даних у двовимірний масив

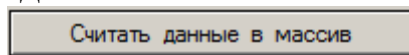
### Демонстраційний приклад 2

Доповнити попередню модель можливістю зчитування введених даних бланка успішності групи в двовимірний масив, виконати тестування (рисунок 2.13).

Для реалізації моделі:

1. Резервуємо місце в оперативній пам'яті під двовимірний масив, що містить 4 стовпця та 7 рядків.

2. Додамо кнопку:



3. У вкладеному циклічному процесі за рядками й стовпцями зробимо зчитування в двовимірний масив сітки даних.

Додамо в лістинг попередньої програми наступний код:

```
Sub Считать_в_массив_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Считать_в_массив.Click
    ' ФОРМИРУЕМ ВЛОЖЕННЫЙ ЦИКЛИЧЕСКИЙ ПРОЦЕСС:
    For i = 1 To 7
        For j = 1 To 4
            ведомость(i, j) = DataGridView1.Item(j - 1, i - 1).Value
            TextBox1.Text = TextBox1.Text & " " & ведомость(i, j)
        Next
        TextBox1.Text = TextBox1.Text & vbNewLine
    Next
End Sub
```

При запуску програми заповнимо сітку даних довільними значеннями.



Зауваження: зверніть увагу, що нумерація рядків і стовпців у DataGridView починається з нуля.

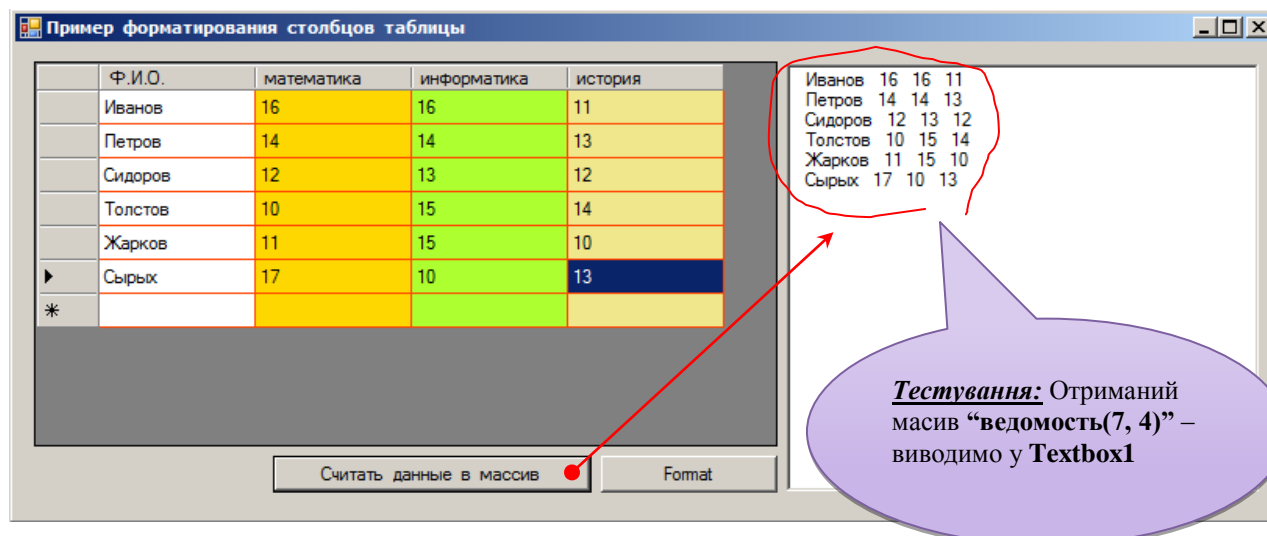


Рисунок 2.13 – Результат зчитування даних з DataGridView у двовимірний масив

### Обробка масиву даних

#### Демонстраційний приклад 3

Доповнити попередню модель можливістю пошуку максимальної (мінімальної) оцінки стовпця «история».

Доповнимо вищенаведені лістинги програм наступним кодом:

---

```
Dim строка As Integer
  Dim Столбец_история(6) As Double
  For строка = 0 To 6
    Столбец_история(строка) = DataGridView1.Item(3, строка).Value
  Next
  MsgBox("Максимальная оценка по истории = " & Столбец_история.Max & _
  Chr(13) & "Минимальная оценка по истории = " & Столбец_история.Min)
```

---

У даному лістингу програми ми продемонстрували можливість використання методів «Max» і «Min», які повертають відповідно максимальне й мінімальне значення, що містяться в послідовності значень «Столбец\_история» (рисунок 2.14).

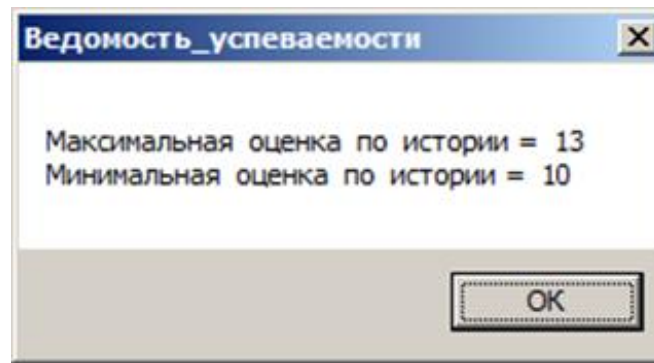


Рисунок 2.14 – Діалогове вікно успішності з історії

Таким чином, ми навчилися:

- форматувати сітку даних;
- ідентифікувати двовимірні масиви;
- зчитувати інформацію із сітки даних у масив;
- обробляти отримані масиви.

## 2.6 Реалізація моделі «Матриця-1»

Умова. Транспортна компанія «GT-Trans» проводить дослідження з оптимізації власної стратегії на ринку надання логістичних послуг. Для цього аналізуються темпи зростання доходів від перевезень вантажів за найбільш популярними маршрутами різних компаній-замовників. Аналітик формує зведену таблицю, що містить показники зростання доходності відповідних маршрутів за поточний тиждень. На підставі цих даних необхідно сформувати пакет перевезень на наступний тиждень, який буде містити найбільш доходні маршрути, крім того, необхідно мати можливість відсіювати варіанти з мінімальним темпом зростання доходності за напрямками слідування вантажів.

У допомогу аналітику необхідно розробити калькулятор для внесення даних та відповідних розрахунків (можливість знаходження максимальних та мінімальних елементів матриці) зі зручним інтерфейсом (рисунки 2.15–2.18).

Матрица\_1(Работа с заданным двумерным массивом) DEN

Найти (max),(min) элементы по строчно для произвольно заданной матрицы любого размера.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & \dots & a_{mn} \end{pmatrix}$$

показники зростання доходності маршрутів за напрямками

показники зростання доходності маршрутів за замовниками

Введи размерность матрицы?

Рисунок 2.15 – Матриця-1

РАЗМЕРНОСТЬ ПО ВЕРТИКАЛИ

Введите количество строк в планируемой таблице

7

OK

Отмена

Рисунок 2.16 – Діалогове вікно введення параметрів масиву

РАЗМЕРНОСТЬ ПО ГОРИЗОНТАЛИ

Введите количество столбцов в планируемой таблице

6

OK

Отмена

Рисунок 2.17 – Діалогове вікно «Розмірність по горизонталі»

Найти (max),(min) элементы по строчно для произвольно заданной матрицы любого размера.

№ П/П	№1	№2	№3	№4	№5	№6	max(строки)
стр.№1	1	2	35	4	21	5	35
стр.№2	-62	75	56	45	23	4	75
стр.№3	19	74	6	45	-20	8	74
стр.№4	73	22	15	-6	-30	96	96
стр.№5	68	11	2	42	57	1	68
стр.№6	43	13	41	24	76	10	76
стр.№7	10	33	7	8	9	44	44

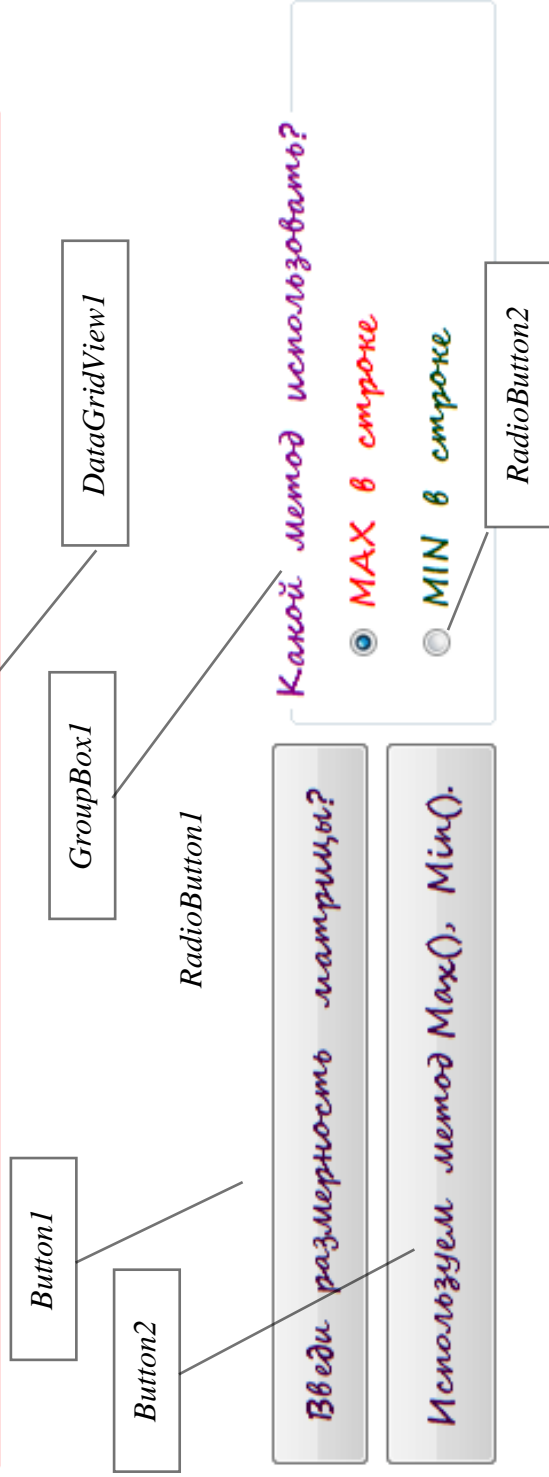


Рисунок 2.18 – Матрица-1

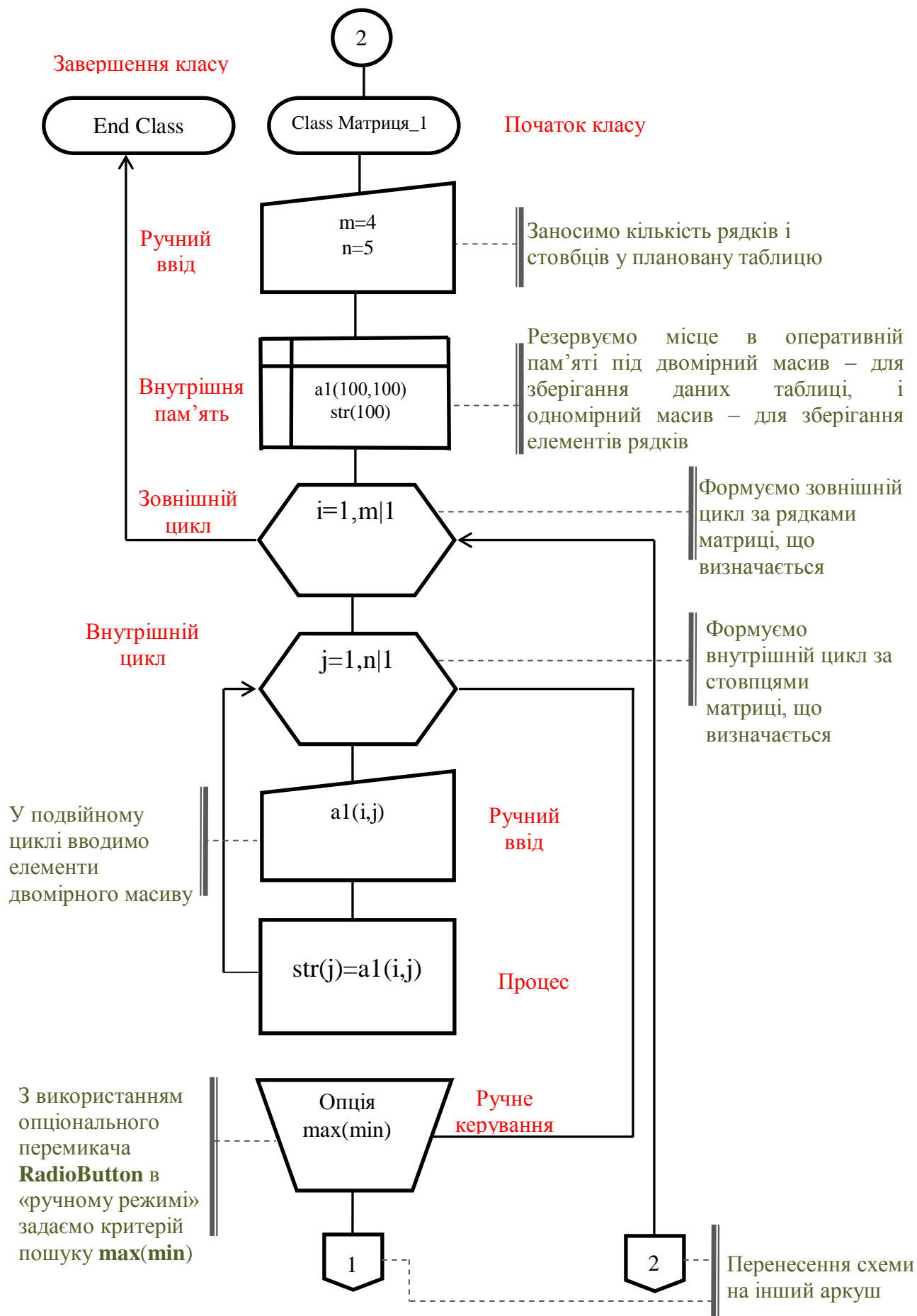


Рисунок 2.19 – Блок-схема програми моделі «Матриця-1»

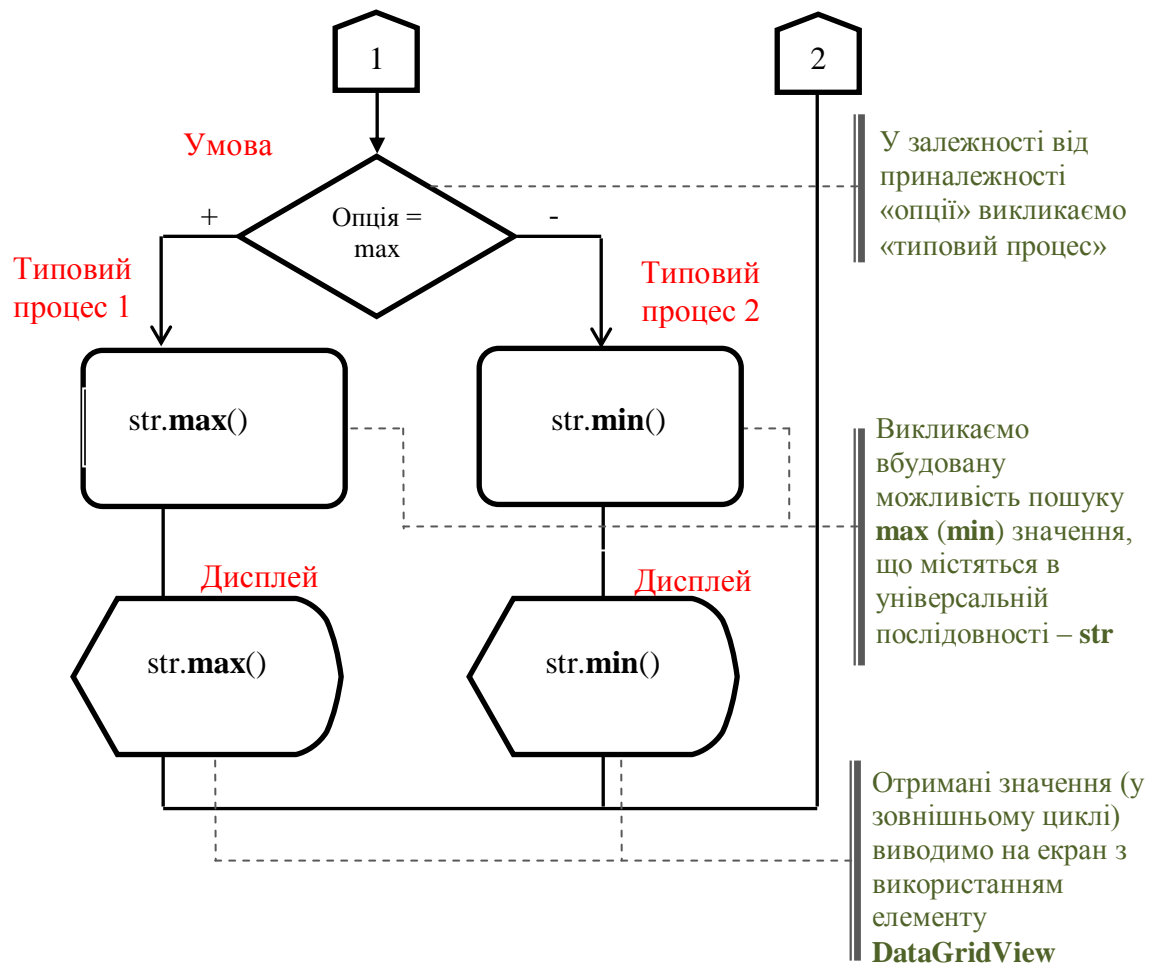


Рисунок 2.19, аркуш 2

## Лістинг програми моделі «Матриця-1»

```

Public Class Матриця_1
    '<Идентифицируем все элементы которые участвуют в последующем коде: >
    Dim i As Integer, j As Integer, m As Integer = 4, n As Integer = 5
    'Dim a1(100, 100) As Nullable(Of Double)'(ЕСЛИ НАДО) Nullable – класс
    ' который поддерживает тип значения, _
    ' которому может быть присвоено значение Nothing, (где Nothing-представляет
    ' значение по умолчанию любого типа данных.
    Dim a1(100, 100) As Double : Dim str(100) 'Резервируем место в
    ' оперативной памяти под двумерный a1(100, 100)_
    ' массив - для хранения данных таблицы и одномерный str(100) массив
    ' для хранения элементов строк.
    Dim format As New DataGridViewCellStyle 'DataGridViewCellStyle-Представляет
    ' сведения о форматировании и стиле, _
    ' применяемые к отдельным ячейкам в элементе управления DataGridView.
    #Region "В данном блоке выполним визуальное оформление DataGridView _
    (БОЛЕЕ ПОДРОБНОЕ ИЗУЧЕНИЕ ПРОГРАММИРОВАНИЯ DataGridView)"
    'ЗАМЕЧАНИЕ:Код значительно сократится, если все прописать в окне свойств
    Private Sub Button1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button1.Click
        m = InputBox("Введите количество строк в планируемой таблице", _
            "РАЗМЕРНОСТЬ ПО ВЕРТИКАЛИ", 4)
    
```

```

n = InputBox("Введите количество столбцов в планируемой таблице", _
            "РАЗМЕРНОСТЬ ПО ГОРИЗОНТАЛИ", 5)
With format 'Определяем, как будет определено форматирование ячеек в
            'DataGridView.
    .BackColor = Color.Black
    .ForeColor = Color.FromArgb(225, 225, 255, 1)
End With
With DataGridView1
'Выполняем блок операторов с повторяющимися ссылками на DataGridView, т.е.:
.Columns.Clear() 'Clear- Очищает коллекцию, в данном случае - столбцы
.Rows.Clear() 'Clear- Очищает коллекцию, в данном случае - строки
.RowHeadersVisible = False 'RowHeadersVisible-указываем, отображается
                            'ли столбец, содержащий заголовки строк.
.AllowUserToAddRows = False 'AllowUserToAddRows-указываем,
                            'отображается ли для пользователя параметр добавления строк.
.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells
'AutoSizeColumnsMode-Возвращает или задает значение, указывающее,
'как определяется ширина столбца.
.ColumnCount = n + 1 '<Ru.Count-Граф>Возвращает или задает число
'столбцов, отображаемых в объекте DataGridView.
.RowCount = m + 1 '<Ru.Count-Граф>Возвращает или задает число
'строк, отображаемых в объекте DataGridView.
.Rows(0).Visible = False 'DataGridView.Rows-Получает коллекцию,
'содержащую все строки в элементе управления DataGridView.
.GridColor = Color.Red 'Возвращает или задает цвет линий сетки,
'разделяющих ячейки объекта DataGridView.
.BackgroundColor = Color.White 'Возвращает или задает цвет фона для
' объекта DataGridView.
.ForeColor = Color.Blue 'Знаем
.Font = New System.Drawing.Font("Times New Roman", 14) 'Знаем
.DefaultCellStyle.BackColor = Color.BlanchedAlmond
'Возвращает или задает стиль ячейки по умолчанию, который будет
'применяться к ячейкам в объекте DataGridView, если не заданы
'какие-либо другие свойства стиля ячейки.
.Columns(0).ReadOnly = True 'DataGridView.Columns Получает
'коллекцию, содержащую все столбцы в элементе управления,
' в данном случае нулевой столбец .ReadOnly- задает значение,
'указывающее, может ли пользователь изменять ячейки.
.Rows(1).Cells(1).ToolTipText = "Вводите только числовые значения"
'DataGridViewRow.Cells - возвращает коллекцию ячеек,
'заполняющих строку.
'.DefaultCellStyle.NullValue = "0"'(если нужно)
'DataGridViewCellStyle.NullValue
'-Получает или задает отображаемое значение ячейки DataGridView.
'
For j = 0 To n
' DataGridView1.Item(j, 0).Value = "стол.№" & j:'Если требуется записать
' шапку столбцов в нулевой строке
.Columns(j).Name = "№" & j ' Если требуется пронумеровать строку-заголовок
Next j
.Columns(0).Name = "№ п/п"
.Columns.Add("Имя_столбца", "надпись")
.Columns("Имя_столбца").DefaultCellStyle = format
'где [объект].DefaultCellStyle- возвращает или задает стиль по
' умолчанию для ячеек = format заданный выше.
.Columns("Имя_столбца").ReadOnly = True

```

```

.Columns("Имя_столбца").Visible = False
For i = 0 To m
    .Item(0, i).Value = "стр.№" & i
    'Записываем шапку строк в нулевом столбце
    'ЗАМЕЧАНИЕ: [.Item( _ , _ )]Предоставляет индекатор для
    'получения или задания ячейки, расположенной на пересечении
    'указанных [СТОЛБЦА] и [СТРОКИ] (НЕ наоборот-не перепутай)
Next i
.Columns(0).Name = "№ п/п"
End With

Button2.Visible = True : GroupBox1.Visible = True : PictureBox1.Visible = False

MsgBox("Заполни таблицу числовыми значениями", MsgBoxStyle.Information, _
    "правильно выбирай разделитель")

End Sub
#End Region

#Region " Поиск макс. или мин. элемента с использованием метода [max()] _
или [min()](НЕПОСРЕДСТВЕННОЕ РЕШЕНИЕ МОДЕЛИ)"
Private Sub Button2_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button2.Click
On Error GoTo 999 'Обрабатываем НЕ числовой ввод данных в DataGridView.
    Проверочная_для_mat1.Show() : Проверочная_для_mat1.Activate()
    'Показываем и активирует форму,переводит на нее фокус.
    Проверочная_для_mat1.TextBox1.Text = "Проверка считывания данных с _
таблицы в массив: " & Chr(13) & _
    "В ОПЕРАТИВНОЙ ПАМЯТИ ПОЛУЧИЛИ СЛЕДУЮЩИЙ ДВУМЕРНЫЙ МАССИВ A(m,n):" _
    & vbCrLf
    '< Следующий двойной цикл отвечает за считывание элементов с
    ' таблицы в массив a1(i, j): >
With DataGridView1
    .Columns("Имя_столбца").Visible = True 'см.выше
    For i = 1 To .Rows.Count - 1 'Получает число строк в коллекции.
        For j = 1 To n
            a1(i, j) = CDb1(DataGridView1.Item(j, i).Value) : str(j) = a1(i, j)
            Проверочная_для_mat1.TextBox1.Text = _
            Проверочная_для_mat1.TextBox1.Text & " " & _
            a1(i, j)
            'Покажем упрощенный вывод введенного двумерного массива в
            'многострочное текстовое поле [.TextBox1]
        Next j
        Проверочная_для_mat1.TextBox1.Text = _
        Проверочная_для_mat1.TextBox1.Text & vbCrLf
        If RadioButton1.Checked = True Then
            DataGridView1.Item(n + 1, i).Value = str.Max()
            'где Max()-возвращает максимальное значение_
            'содержащееся в универсальной последовательности.
            .Columns("Имя_столбца").HeaderText = "max(строки)"
            'DataGridViewColumn.HeaderText - свойство которое возвращает или _
            ' задает текст ячейки заголовка столбца.
        Else
            DataGridView1.Item(n + 1, i).Value = str.Min()
            'где Min()-возвращает максимальное значение_
            'содержащееся в универсальной последовательности.

```



```

        .Columns("Имя_столбца").HeaderText = "min(строки)"
    End If
Next i
End With
Exit Sub

999:MsgBox("ПОЛЬЗОВАТЕЛЬ!" & Chr(13) & "В исходной матрице введены НЕ _
        числовые значения" & Chr(13) & _
        "ПОВТОРИ ВВОД ЧИСЛОВЫХ ЗНАЧЕНИЙ В ТАБЛИЦЕ", _
        MsgBoxStyle.Exclamation, "ОШИБКА ИДЕНТИФИКАЦИИ ВВЕДЕННЫХ ДАННЫХ")
End Sub
#End Region
End Class


```

## 2.7 Діалогові вікна роботи із зовнішніми файлами

Дуже часто при реалізації техніко-економічних моделей доводиться зустрічатися з необхідністю обробки великих масивів даних, які потрібно зберігати на зовнішні носії з наступним використанням. Для цього нам необхідно вивчити наступні можливості Visual Studio:

1. Запис даних у зовнішні файли з наступним використанням.
2. Використання діалогових вікон збереження й відкриття файлів.

### Діалогове вікно SaveFileDialog

 **SaveFileDialog** – клас; запитує в користувача місце розташування для збереження файла (рисунок 2.20).

WriteAllText – метод, що записує текст у файл.

Синтаксис запису:

```
Public Sub WriteAllText(ByVal file As String, ByVal text As String, _
    ByVal append As Boolean),
```

де file – файл, у який необхідно виконати запис;

text – текст, який необхідно записати у файл;

append – специфікатор: True для додавання вмісту файла; False для перезапису вмісту файла. (Значення за замовчуванням – False.)

У якості параметра file можна використовувати ім'я файла діалогового вікна SaveFileDialog. Таким чином, для запису тексту в зовнішній файл, можна використовувати наступну конструкцію:

```
My.Computer.FileSystem.WriteAlltext(SaveFileDialog1.FileName, [текст], False)
```

### Демонстраційний приклад 4

Записати в зовнішній файл таблицю табулювання функції  $y = \sin(x)$  на інтервалі  $[-1,1]$  з довільно обраним кроком. У графічному інтерфейсі

передбачити діалогове вікно вибору імені й точки збереження файла даних.

Реалізація:

- 1) додамо на форму елемент керування SaveFileDialog1;
- 2) додамо керуючу кнопку й закріпимо за нею наступний код:

---

```
Dim x As Double, шаг_табулирование As Double = 0.5
If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
' проверка истинности заполнения диалогового окна SaveFileDialog
  My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, _
    "Табулирование y=sin(x)", False)
    ' название таблицы (шапка таблицы)
  For x = -1 To 1 Step шаг_табулирование ' циклический процесс _
    от [-1] до [1] с шагом [шаг_табулирование=0.5]
    Dim y As Double = Math.Sin(x)
    My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, vbNewLine _
      & "Y(" & x & ")=" & y, True)
      ' вызов диалогового окна сохранения файла и запись _
        значений табулирования функции
  Next
End If ' закрытие блока IF
```

---

Після запуску програми та заповнення поля імені файла, отриманий файл відкриємо будь-яким текстовим редактором (рисунок 2.21).

### Діалогове вікно OpenFileDialog

 **OpenFileDialog** – клас: запитує користувача про відкриття файла (рисунок 2.22).

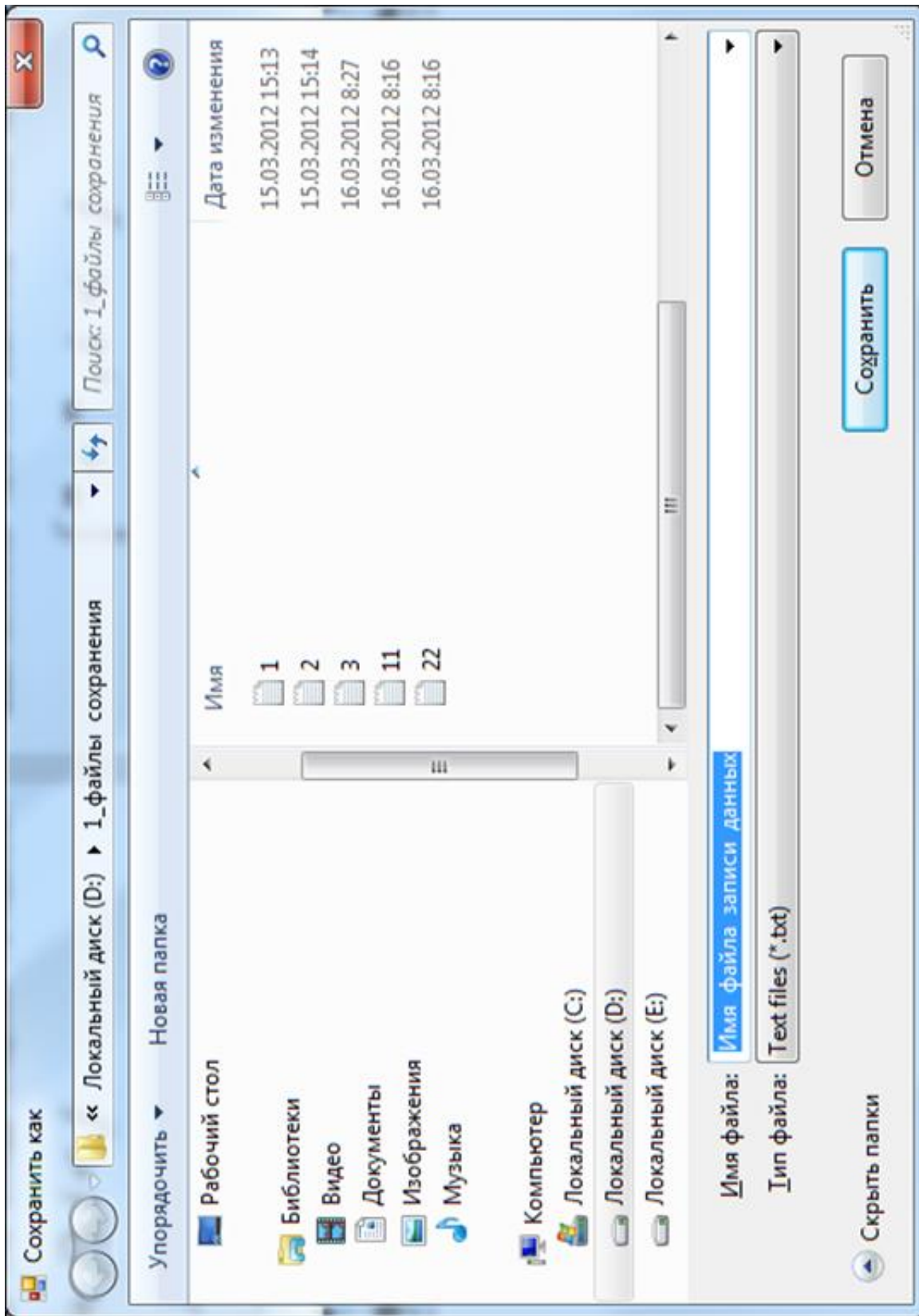


Рисунок 2.20 – Диалоговое окно SaveFileDialog

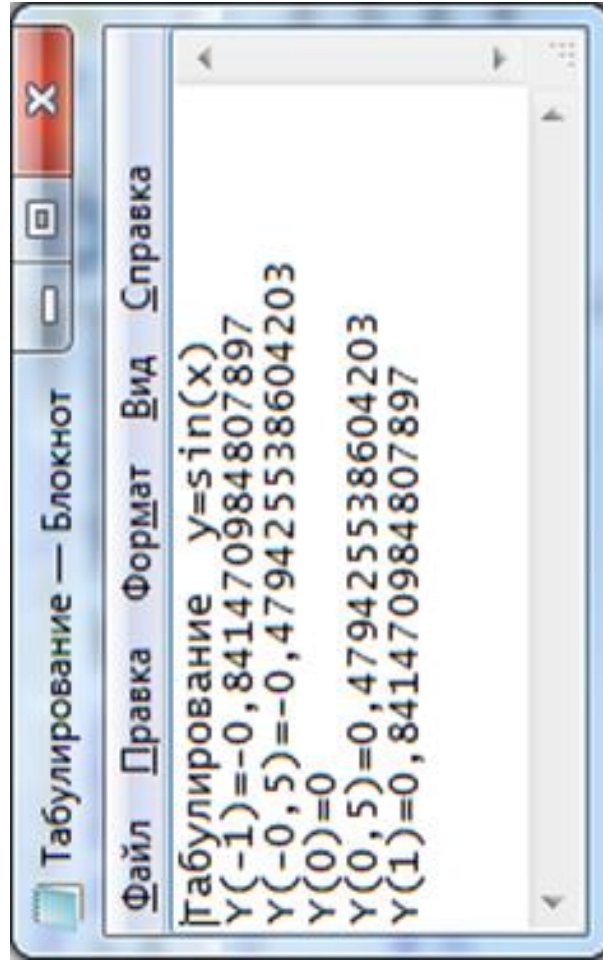


Рисунок 2.21 – Відкриття файла збережених даних

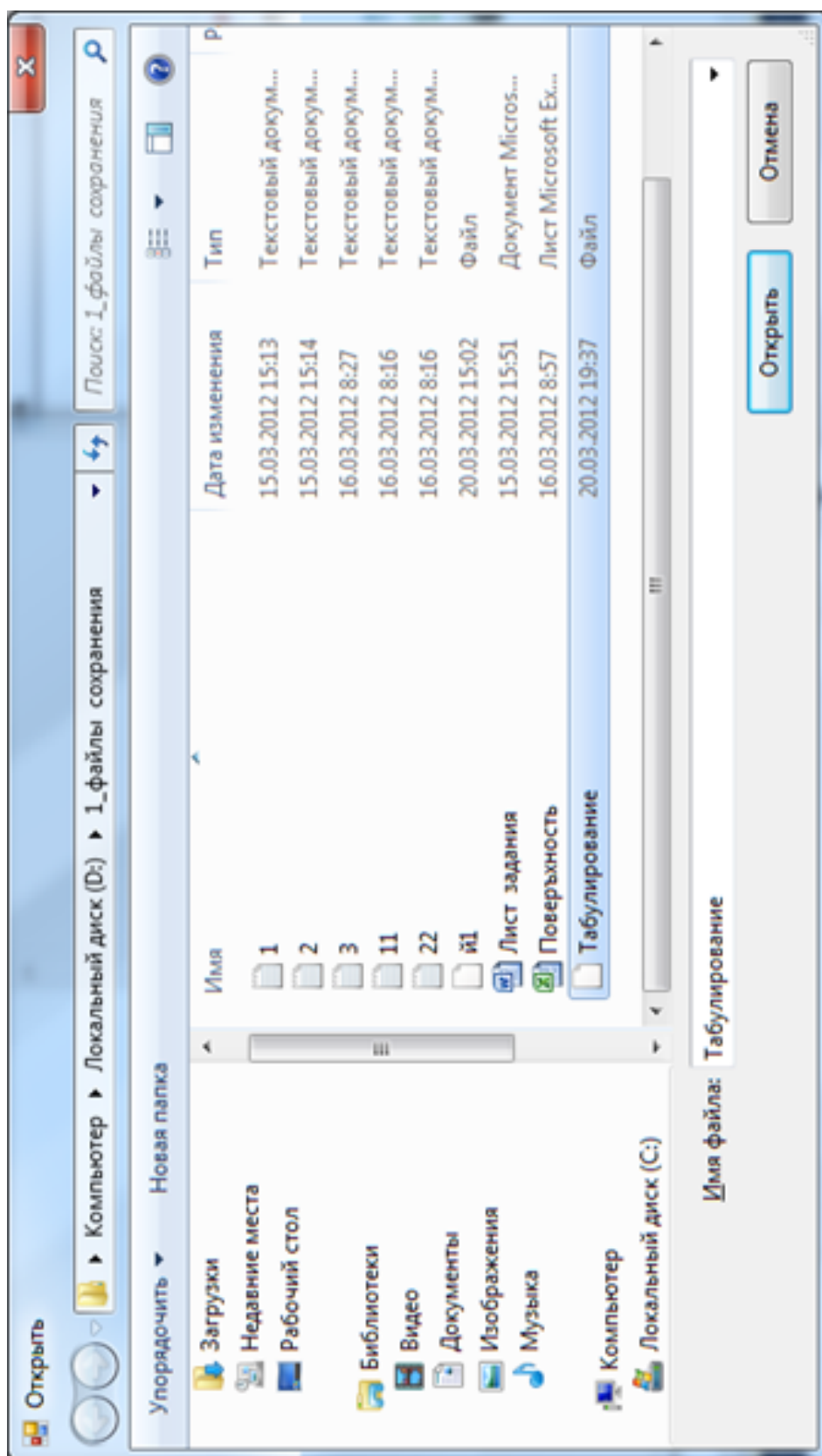


Рисунок 2.22 – Диалогове вікно OpenFileDialog

## Демонстраційний приклад 5

Доповнити «Демонстраційний приклад 4» можливістю зручного інтерфейсу відкриття збережених даних.

Реалізація:

- 1) додамо на форму елемент керування OpenFileDialog1;
- 2) додамо керуючу кнопку й закріпимо за нею наступний код та одержимо результат (рисунок 2.23).

---

```
If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
    MsgBox(My.Computer.FileSystem.ReadAllText(OpenFileDialog1.FileName), , _
        "Сохранённые данные")
End If
```

---

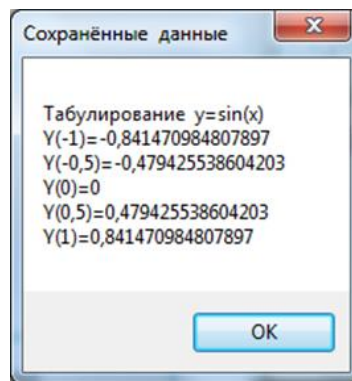


Рисунок 2.23 – Зчитування тексту збережених даних, що записані раніше

Тепер ми можемо перейти безпосередньо до реалізації моделі «Матриця-2».

Зауваження. Модель «Матриця-2» відрізняється від моделі «Матриця-1» тим, що масив даних у моделі «Матриця-1» задавав сам користувач, а масив даних «Матриця-2» будується безпосередньо за заданим правилом.

### 2.8 Реалізація моделі «Матриця-2»

Умова. Транспортна компанія «Macro Trans», проводячи реструктуризацію виробництва, має на меті збалансувати виробничий процес за показниками фондоозброєності. З цією метою було побудовано економетричну модель залежності темпів зростання обсягів надання транспортних послуг від показників темпів зростання кількості робочих ( $i$ ) та обсягу основних фондів ( $j$ ):

$$a(i, j) = \begin{cases} \sin(i + j), i < j & i = \overline{0, n-1} \\ i + j, i = j & \\ \cos(i \cdot j), i > j & j = \overline{0, m-1} \end{cases}$$

Очевидно, що підвищувати темпи зростання обсягів наданих послуг можливо завдяки підвищенню темпів зростання кількості робочих ( $i$ ) і обсягу основних фондів ( $j$ ), причому необхідно забезпечувати баланс зростання цих показників, що забезпечить максимально ефективне використання робочої сили та основних засобів виробництва (рисунок 2.24).

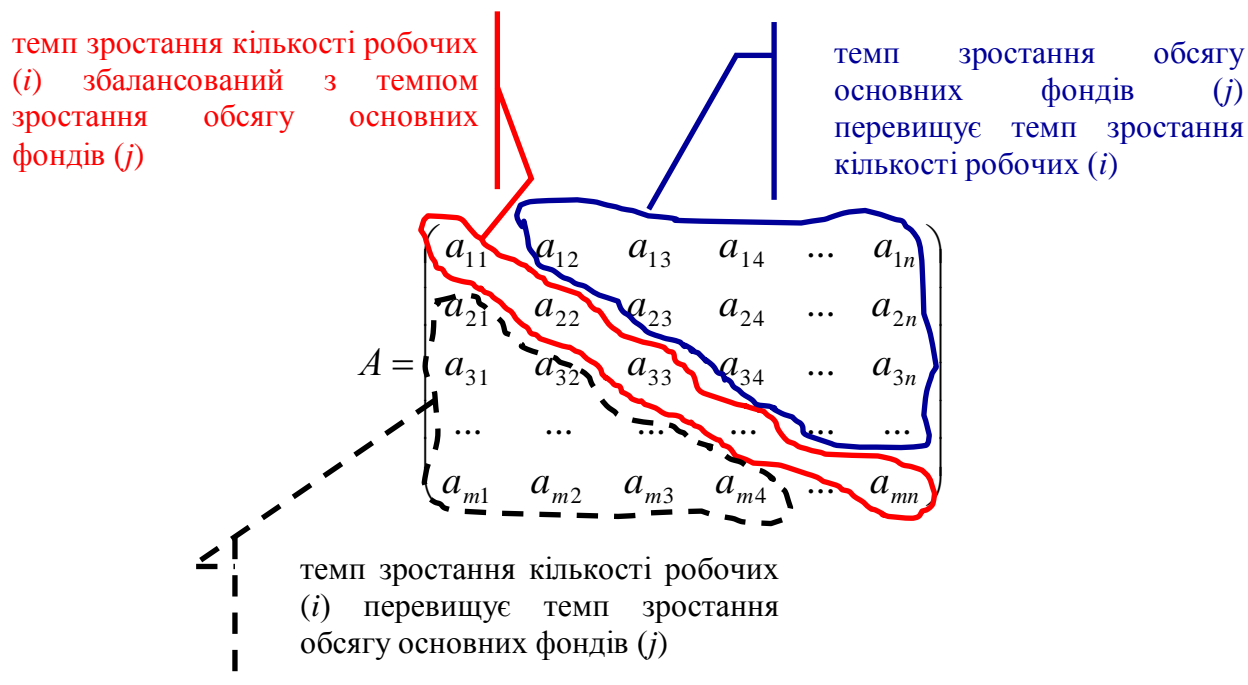


Рисунок 2.24 – Модель збалансування показників виробництва

Згідно з побудованою економетричною моделлю, обчислити матрицю показників темпів зростання обсягів надання послуг.

Для подальшої обробки отриманих даних в аналітичному відділі використовуються різноманітні програмні інструменти підтримки прийняття рішень (рисунок 2.25). Тому з метою забезпечення можливості введення вихідних даних у ці програмні комплекси необхідно організувати збереження отриманого масиву розрахункових даних у зовнішній файл (рисунок 2.26–2.27). Блок-схему алгоритму подано на рисунку 2.28.

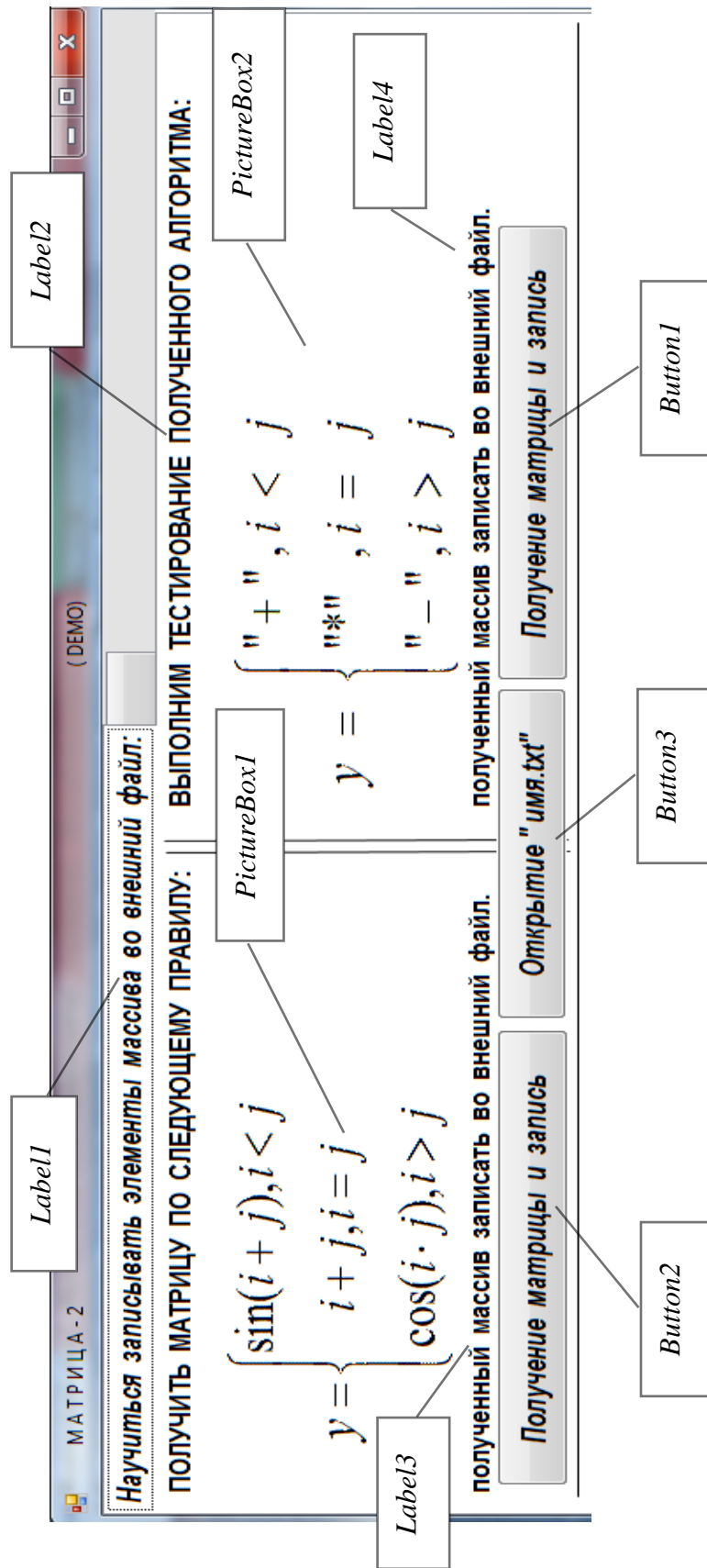


Рисунок 2.25 – Интерфейс користувача моделі «Матриця-2»



DataGridView

0	0	0	0	0	0	0	0	0	0
0,841	2	0,909	0,141	-0,757	-0,959	-0,279	0,657	0,989	0,412
0,909	0,141	4	-0,279	0,989	-0,544	-0,537	0,991	-0,288	-0,751
0,141	-0,757	-0,959	6	-0,537	0,65	-0,751	0,837	-0,906	0,956
-0,757	-0,959	-0,279	0,657	8	0,913	-0,906	0,271	0,551	-0,992
-0,959	-0,279	0,657	0,989	0,412	10	-0,988	-0,428	0,745	0,851
-0,279	0,657	0,989	0,412	-0,544	-1	12	-0,917	-0,768	-0,559
0,657	0,989	0,412	-0,544	-1	-0,537	0,42	14	-0,522	0,167
0,989	0,412	-0,544	-1	-0,537	0,42	0,991	0,65	16	0,254
0,412	-0,544	-1	-0,537	0,42	0,991	0,65	-0,288	-0,961	18

Рисунок 2.26 – Очікувані результати побудови матриці за заданим правилом

*	+	+	+	+	+	+	+	+	+
-	*	+	+	+	+	+	+	+	+
-	-	*	+	+	+	+	+	+	+
-	-	-	*	+	+	+	+	+	+
-	-	-	-	*	+	+	+	+	+
-	-	-	-	-	*	+	+	+	+
-	-	-	-	-	-	*	+	+	+
-	-	-	-	-	-	-	*	+	+
-	-	-	-	-	-	-	-	*	+
-	-	-	-	-	-	-	-	-	*

Рисунок 2.27 – Очікувані результати тестування моделі

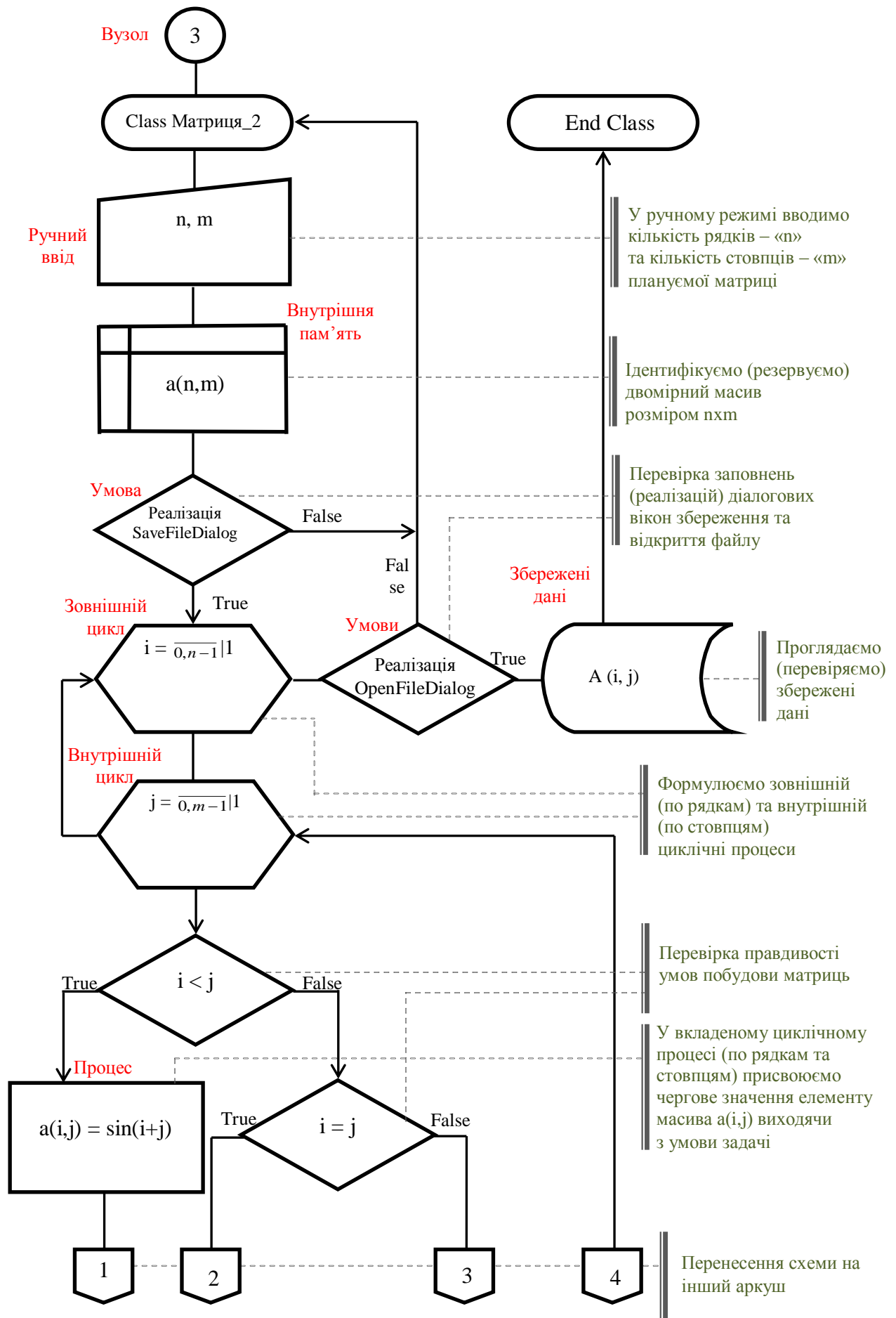


Рисунок 2.28 – Блок-схема моделі «Матриця-2»

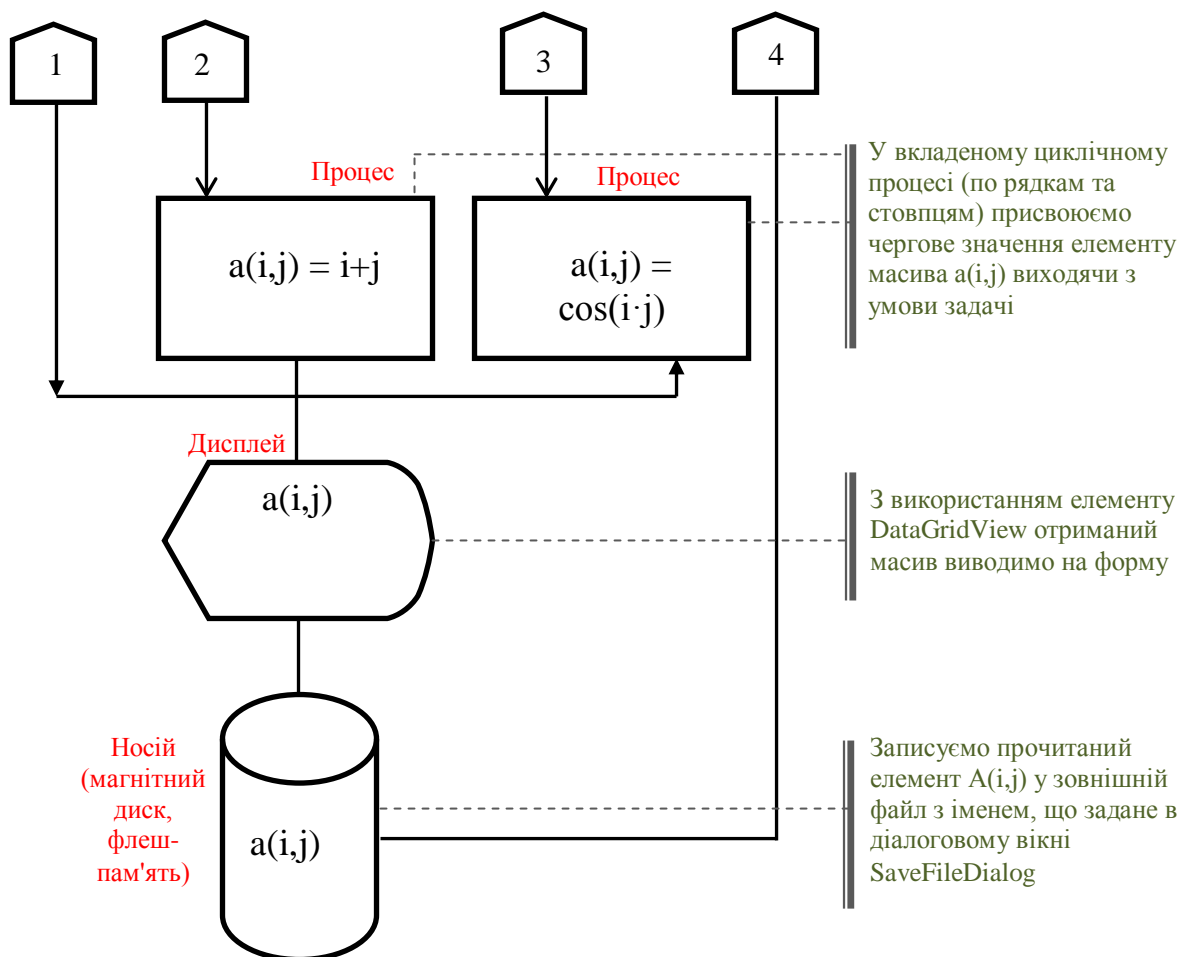


Рисунок 2.28 – Аркуш 2

## Лістинг програми моделі «Матриця-2»

```
Imports System.Math
Public Class Матриця_2
    Dim m As Integer, n As Integer
    #Region "Получение матрицы по заданному правилу"
    Private Sub Button2_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles Button2.Click
        ' My.Computer.FileSystem.WriteAllText("C:\Test.txt", "", False)
        'Используй если просто без SaveFileDialog1
        m = InputBox("Введите количество столбцов матрицы", _
            "РАЗМЕРНОСТЬ ПЛАНИРУЕМОЙ ТАБЛИЦЫ", 14)
        n = InputBox("Введите количество строк матрицы", _
            "РАЗМЕРНОСТЬ ПЛАНИРУЕМОЙ ТАБЛИЦЫ", 20)

        Dim a(n, m) As Double
    If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
        My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, _
            vbCrLf, False)
        With DataGridView1
            .ColumnCount = m ' <Ru.Count-Граф>Возвращает или задает
                'число столбцов,
                'отображаемых в объекте DataGridView.
            .RowCount = n ' <Ru.Count-Граф>Возвращает или задает
                'число строк,
```

```

        'отображаемых в объекте DataGridView.
.GridColor = Color.Red 'Возвращает или задает цвет линий сетки,
        'разделяющих ячейки объекта DataGridView.
.BackgroundColor = Color.White
    'Возвращает или задает цвет фона для объекта DataGridView.
.ForeColor = Color.Blue 'Знаем
.Font = New System.Drawing.Font("Times New Roman", 16) 'Знаем
For i = 0 To n - 1 'Нумерация любого массива начинается с 0
    For j = 0 To m - 1
        If i > j Then 'DataGridView.Item - свойство, которое
            'предоставляет индекатор для получения или
            'задания ячейки, расположенной на пересечении указанных
            'столбца и строки (НЕ НАОБОРОТ)
            .Item(j, i).Value = Round(Sin(i + j), 3)
            a(i, j) = Round(Sin(i + j), 5)
            ' MsgBox(.CurrentCell.Value)
        ElseIf i = j Then
            .Item(j, i).Value = Round(i + j, 3)
            a(i, j) = Round(i + j, 5)
        Else
            .Item(j, i).Value = Round(Sin(i * j), 3) : a(i, j) = _
                Round(Sin(i * j), 5)
        End If

        My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, _
            CStr(a(i, j)) & vbTab, True)
    Next
    My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, _
        vbCrLf, True)
Next
End With
End If
End Sub
#End Region
#Region "Копия предыдущего кода-///ВСТАВЬ ЗНАЧЕНИЯ СВОЕГО ЛИСТА ЗАДАНИЯ///"
Private Sub Button1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
    'My.Computer.FileSystem.WriteAllText("C:\Test.txt", "", False)
    'Используй, если просто без SaveFileDialog1
    m = InputBox("Введите количество столбцов матрицы", , 10)
    n = InputBox("Введите количество строк матрицы", , 10)
    Dim a(n, m)
    If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
        My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, _
            vbCrLf, False)
        With DataGridView1
            .ColumnCount = m '<Ru.Count-Граф>Возвращает или задает
                'число столбцов, отображаемых в объекте DataGridView.
            .RowCount = n '<Ru.Count-Граф>Возвращает или задает
                'число строк, отображаемых в объекте DataGridView.
            .GridColor = Color.Red 'Возвращает или задает цвет линий сетки,
                ' разделяющих ячейки объекта DataGridView.
            .BackgroundColor = Color.White 'Возвращает или задает цвет фона
                'для объекта DataGridView.
            .ForeColor = Color.Blue 'Знаем
            .Font = New System.Drawing.Font("Times New Roman", 16) 'Знаем
        End With
    End If
End Sub

```

```

For i = 0 To n - 1 'Нумерация любого массива начинается с 0
  For j = 0 To m - 1
    If i > j Then 'DataGridView.Item - свойство, которое
      'предоставляет индекатор для получения или _
      'задания ячейки, расположенной на пересечении
      'указанных столбца и строки (НЕ НАОБОРОТ)
      .Item(j, i).Value = "-" : a(i, j) = "- "
      ' MsgBox(.CurrentCell.Value)
    ElseIf i = j Then
      .Item(j, i).Value = "*" : a(i, j) = "* "
    Else
      .Item(j, i).Value = "+" : a(i, j) = "+ "
    End If
  Next j
Next i

My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, _
  CStr(a(i, j)) & vbTab, True)
Next j

My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, vbCrLf, True)
Next i
Dim fileReader As String
fileReader = My.Computer.FileSystem.ReadAllText(SaveFileDialog1.FileName)
'FileSystemProxy.ReadAllText - метод, который возвращает содержимое
' текстового файла как String.
MsgBox(fileReader)
End With
End If
End Sub
#End Region
#Region "Данные элементы стандартные для любого диалогового окна _
SaveFileDialog и DataGridView(просто скопируй)"
Private Sub Form1_Paint(ByVal sender As Object, _
  ByVal e As System.Windows.Forms.PaintEventArgs) Handles Me.Paint
  With SaveFileDialog1
    ' Добавьте расширение по умолчанию, если пользователь забыл
    'добавить расширение.
    'По умолчанию: True.
    .AddExtension = True
    'Если пользователь не указывает расширение и если
    'свойство AddExtension имеет
    'значение True, используйте это расширение. По умолчанию: "".
    .DefaultExt = ".txt"
    '.DefaultExt = ".xls"
    'По умолчанию: "".
    .Filter = _
      "Text files (*.txt)|*.txt|All files|*.*|Excel files (*.xls)|*.xls"
  End With
  '-----
  With OpenFileDialog1
    'Как в VB6, используйте набор пар фильтров,
    'разделенных знаком "|". Каждая
    'пара включает спецификацию описание|файл.
    'Используйте "|" между парами. Не указывайте "|" в конце.
    .Filter = _
      "Text files (*.txt)|*.txt|All files|*.*|Excel files (*.xls)|*.xls"
  End With

```

```
'  
With DataGridView1  
    .RowHeadersVisible = False 'Возвращает или задает значение,  
    'указывающее, отображается ли столбец, содержащий заголовки строк.  
    .ColumnHeadersVisible = False 'Возвращает или задает значение,  
    'указывающее, отображается ли строка заголовков столбцов.  
    'Далее масштабируем строки и столбцы DataGridView1 под  
    'максимальный размер:  
    .AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells  
    'Возвращает или задает значение, указывающее, как определяется  
    'высота строки.  
    .AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.AllCells  
    'Возвращает или задает значение, указывающее, как определяется  
    'ширина столбца.  
End With  
End Sub  
#End Region  
  
#Region "Кнопка отвечающая за открытие текстового файла записанного ранее"  
Private Sub Button3_Click(ByVal sender As System.Object, _  
    ByVal e As System.EventArgs) Handles Button3.Click  
    If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then  
        'My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName,  
        'CStr(a(i, j)) & vbTab, True)  
  
        MsgBox(My.Computer.FileSystem.ReadAllText(OpenFileDialog1.FileName))  
    End If  
End Sub  
#End Region  
End Class
```

---

## 3 ПРОЦЕДУРИ У VB

Процедура є блоком операторів, що об'єднані оператором оголошення (Function, Sub) і відповідного оголошення End. Усі інструкції, що виконуються, повинні перебувати в межах деякої процедури.

### 3.1 Виклик процедури та повернення з неї

Процедура викликається з будь-якої частини програмного коду. Подібна операція відома як виклик процедури. Після завершення процедури вона повертає керування коду, що викликав її, який називається викликаючим кодом. Викликаючий код – це інструкція або вираз у складі інструкції, який визначає процедуру за іменем та передає їй керування.

Процедура повертає керування викликаючому коду після завершення роботи. Керування потім передається викликаючому коду, слідуючи за точкою виклику процедури.

Керування негайно вертається викликаючому коду оператором Return. Інструкції, що слідують за інструкцією Return, не виконуються.

Керування негайно вертається викликаючому коду оператором Exit Sub або Exit Function. Якщо процедура не має операторів Return, Exit, вона завершується оператором End Sub, End Function, що впливають за останнім оператором тіла процедури.

### 3.2 Параметри й аргументи. Типи процедур

У більшості випадків процедурі необхідно працювати з різними даними при кожному її виклику. Можна передати ці відомості в процедуру як частину виклику процедури. Процедура визначає нуль або більше параметрів, кожний з яких представляє значення, що очікується нею для приймання. Кожному параметру у визначенні процедури відповідає аргумент у виклику процедури.

ByRef указує такий спосіб передачі аргументу, при якому викликана процедура може змінити значення змінної, що міститься в аргументі викликаючого коду.

ByVal указує такий спосіб передачі аргументу, при якому викликана процедура не може змінити значення змінної, що міститься в аргументі викликаючого коду.

У нашій роботі ми познайомимося з наступними типами процедур:

1) процедура типу Sub виконує дії, але не повертає значення у викликаючий код;

2) процедура типу Function повертає значення у викликаючий код, також може виконувати інші дії перед поверненням.

### 3.3 Процедури й структурований код

Процедури застосовуються для виконання повторюваних задач або задач, які потрібно виконувати в різних компонентах коду, наприклад, для часто використовуваних розрахункових операцій.

Якщо розділити великі процедури на декілька менших, не порушуючи при цьому загальної структури алгоритму, програмний код буде зручніше читати. Процедури можна викликати з різних частин коду, тому їх можна використовувати в якості елементарних структурних блоків при створенні додатка.

Структурування коду за допомогою процедур надає наступні можливості:

- 1) процедури дозволяють розбити програму на окремі логічні блоки. Налагоджувати такі окремі блоки легше, чим усю програму цілком;
- 2) процедури, що розроблені для однієї програми, можна в тому ж вигляді (або після внесення незначних змін) використовувати в іншій програмі. Це допомагає уникнути дублювання коду.

### 3.4 Створення процедури

Процедура міститься між початковим оператором оголошення (Sub або Function) і кінцевим оператором оголошення (End Sub або End Function). Увесь код процедури міститься між цими операторами. Процедура не може містити іншу процедуру, тому її оператори початку й кінця повинні бути поза будь-якими іншими процедурами. Якщо існує код, що виконує ту саму задачу в різних місцях, можна оформити цю задачу як процедуру, а потім викликати її з різних місць коду.

Створення процедури, що не повертає значення (процедура типу Sub)

Процедура Sub – це послідовність операторів, які містяться між операторами Sub і End Sub. Процедура Sub виконує необхідні дії та повертає контроль коду виклику, але вона не повертає значення в код виклику.

Для створення такої процедури необхідно виконати наступні дії:

- 1) увести ключове слово Sub з іменем процедури, й потім список параметрів у круглих дужках;
- 2) помістити оператори коду процедури між оператором Sub і End Sub.

Нижче наведено приклад кодів виклику процедури типу Sub (рисунок 3.1):



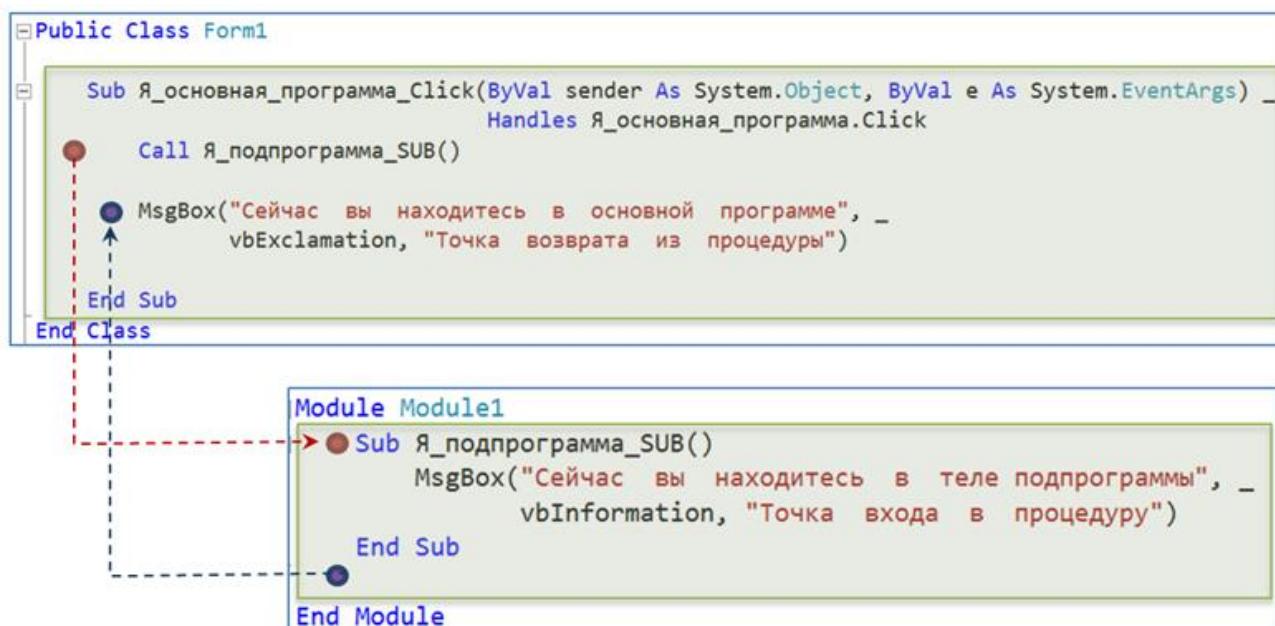


Рисунок 3.1 – Схема виклику процедури типу Sub

Під час запису лістингу процедури необхідно додати в проект програмний модуль (рисунок 3.2):

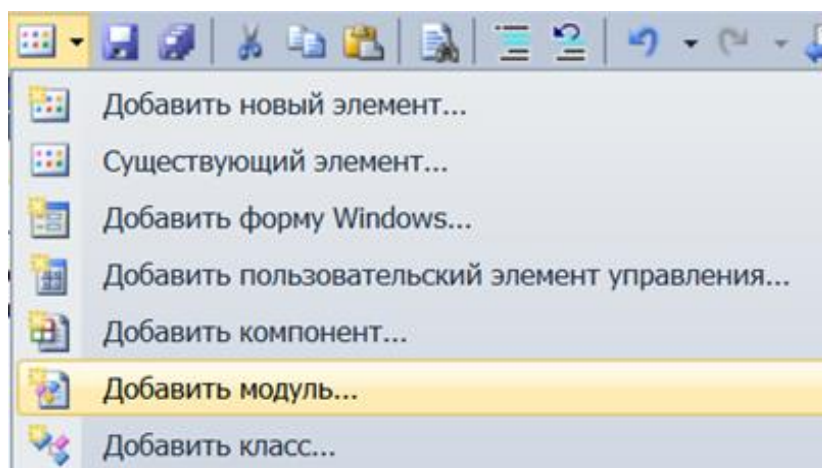


Рисунок 3.2 – Додавання модуля (Module) у проект

Після запуску програми ми можемо відстежити точку входу й виходу з процедури за наступними діалоговими вікнами (рисунок 3.3 та 3.4):

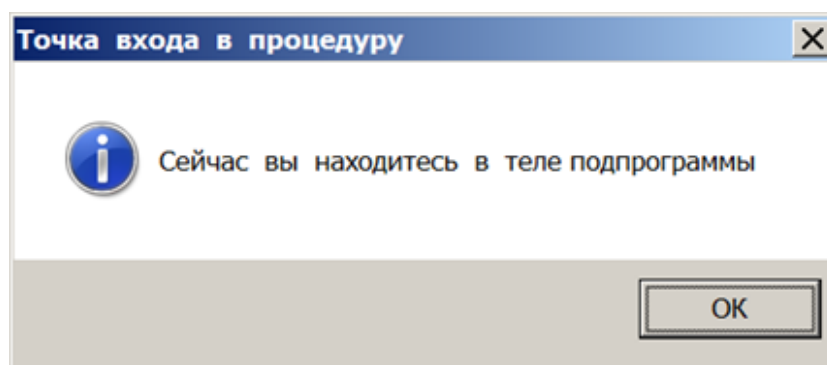


Рисунок 3.3 – Виклик діалогового вікна з тіла процедури

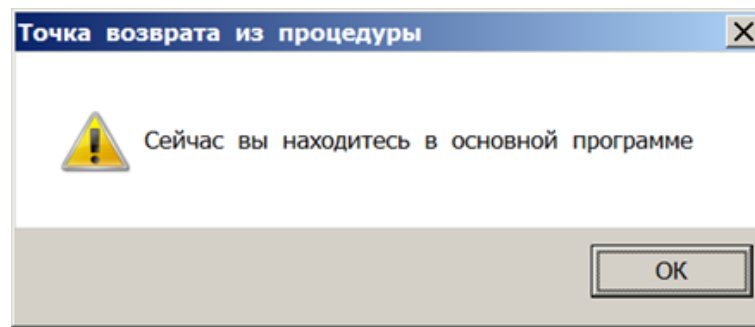


Рисунок 3.4 – Виклик діалогового вікна з основної програми

### Створення процедури, що повертає значення (процедура типу Function)

Процедура Function – це послідовність операторів, які містяться між операторами Function і End Function. Процедура Function виконує задачу та повертає керування коду, що викликав її. Разом із керуванням коду, що викликав функцію, повертається значення.

Для створення такої процедури необхідно виконати наступні дії:

1) після ключового слова Function вказати ім'я процедури, потім список параметрів у круглих дужках, а потім оператор As, що задає тип даних значення, що повертається;

2) помістити оператори коду процедури (тіло процедури) між оператором Function і End Function.

Оператор Return використовується для повернення значення в повертаючий код.

Нижче наведено приклад кодів виклику процедури типу Function, яка прораховує значення площини кола із заданим радіусом (рисунок 3.5 та 3.6):

```
Public Class Form1
    Sub Я_основная_программа1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Я_основная_программа1.Click
        Dim радиус = 2
        MsgBox("Площадь круга с заданным радиусом = " & Площадь_круга(радиус) , vbQuestion, "Вызываем функцию с именем (Площадь_круга)")
    End Sub
End Class

Module Module2
    Function Площадь_круга(ByVal радиус) As Single
        Return Math.PI * радиус ^ 2 'Return-Возвращает управление коду, вызвавшему процедуру Function
    End Function
End Module
```

Рисунок 3.5 – Лістинг основної програми та процедури-функції

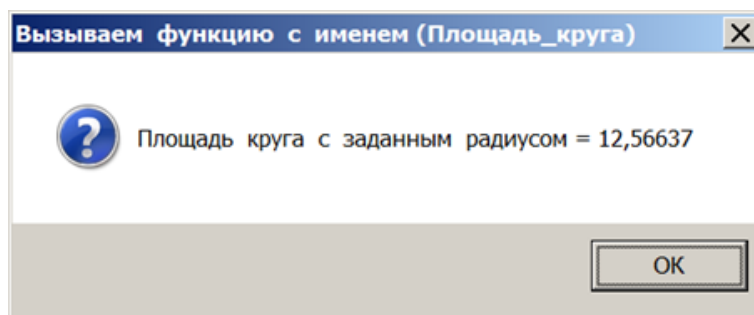


Рисунок 3.6 – Діалогове вікно основної програми з розрахованими даними площі кола в процедурі (Площа\_кола) модуля (Module2)

### 3.5 Реалізація моделі «Процедура – Sub»

Умова. Реалізувати програмний додаток (рисунок 3.8), що використовується для дослідження трифакторної виробничої функції мережі автосервісів, яка має вигляд:

$$U = f(X, Y, Z),$$

де  $U$  – темпи зростання обсягу наданих послуг при даних технологічних факторах;

$X, Y, Z$  – функції темпів зростання витрат капіталу, праці й матеріалів, які мають вигляд:

$$X = K(t)$$

$$Y = L(t)$$

$$Z = M(t)$$

де  $t$  – показник фактора часу; від’ємні значення показника відповідають періодам часу, що передують базовому.

Крім того, вищезгадані функції містять технологічні коефіцієнти:  $a, b, c, m$ , значення яких визначається на основі статистичної інформації та відповідних економетричних моделей.

Під час реалізації моделі доцільно провести візуалізацію результатів розрахунків (рисунок 3.7), побудувавши графіки функцій, наприклад, використовуючи Microsoft Excel (рекомендації до побудови графіків функцій дивіться у [9, с. 6, 16]). Крім того, рекомендується застосувати інструменти прогнозування, які доступні в Microsoft Excel (рекомендації до використання інструментів прогнозування дивіться у [9, с. 47]). Таким чином, був зроблений прогноз темпів зростання обсягу наданих послуг мережі автосервісів ( $U$ ) на 2 кроки вперед, до 11 періоду включно (рисунок 3.7).

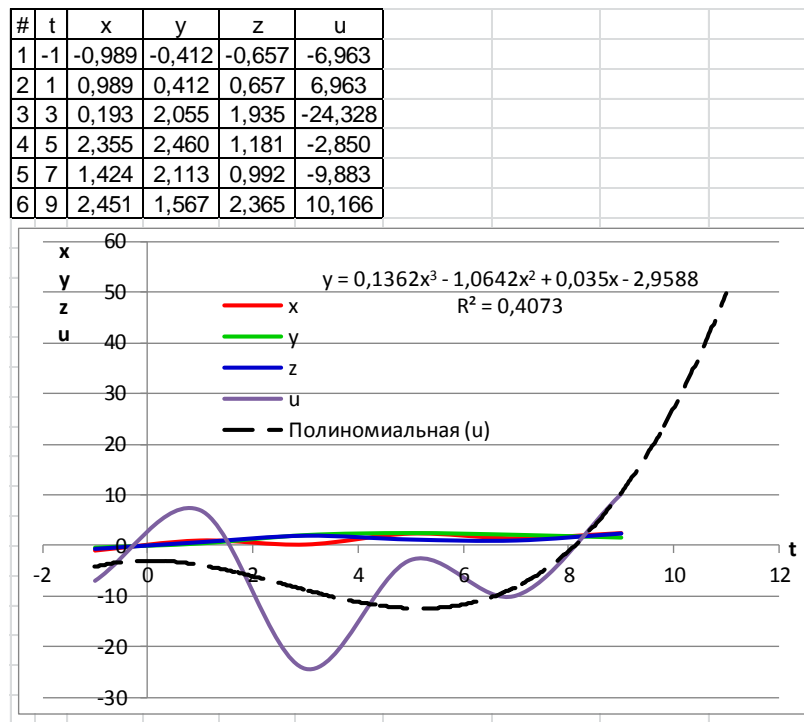


Рисунок 3.7 – Візуалізація розрахунків моделі «Процедура – Sub»

Для подальшої обробки отриманих даних, можуть використовуватися різноманітні програмні інструменти підтримки прийняття рішень. Тому з метою забезпечення можливості введення вихідних даних у ці програмні комплекси необхідно організувати збереження отриманої табличної структури в зовнішній файл (рисунок 3.8–3.9).

Для розрахунку  $X$ ,  $Y$ ,  $Z$  необхідно передбачити підпрограму, яку є можливість у подальшому інкапсулювати у вигляді бібліотеки стандартних модулів та використовувати в якості аналітичного інструменту.

Блок-схему алгоритму подано на рисунках 3.10 та 3.11.

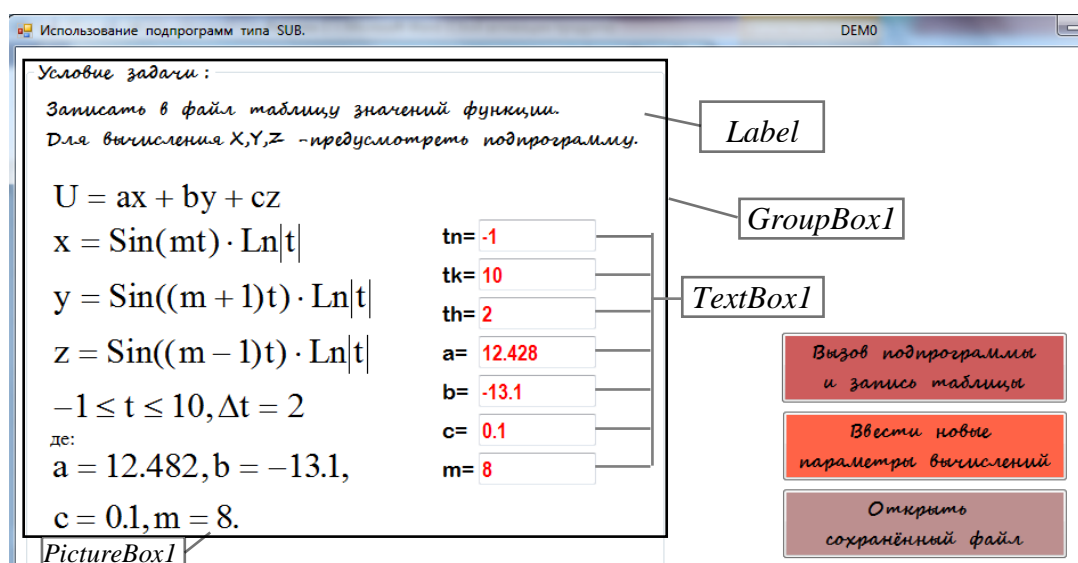


Рисунок 3.8 – Графічний інтерфейс користувача моделі «Процедура – Sub»

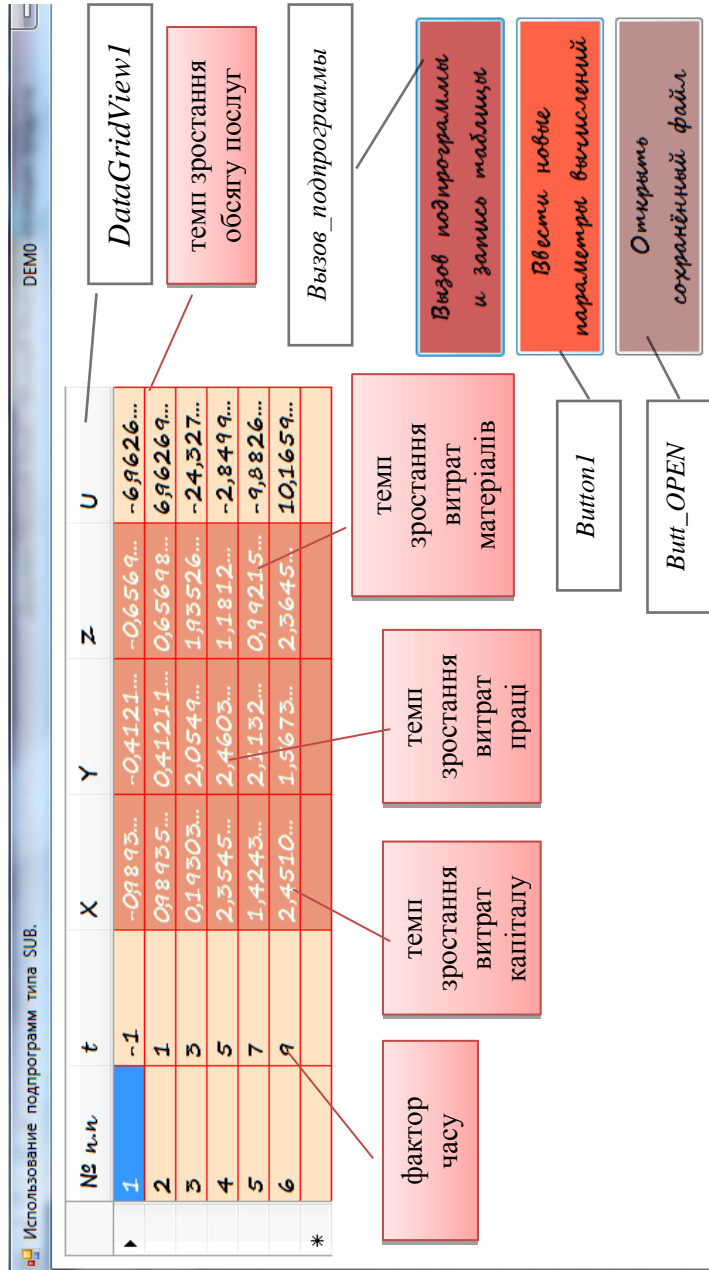


Рисунок 3.9 – Графічний інтерфейс користувача моделі «Процедура – Sub»

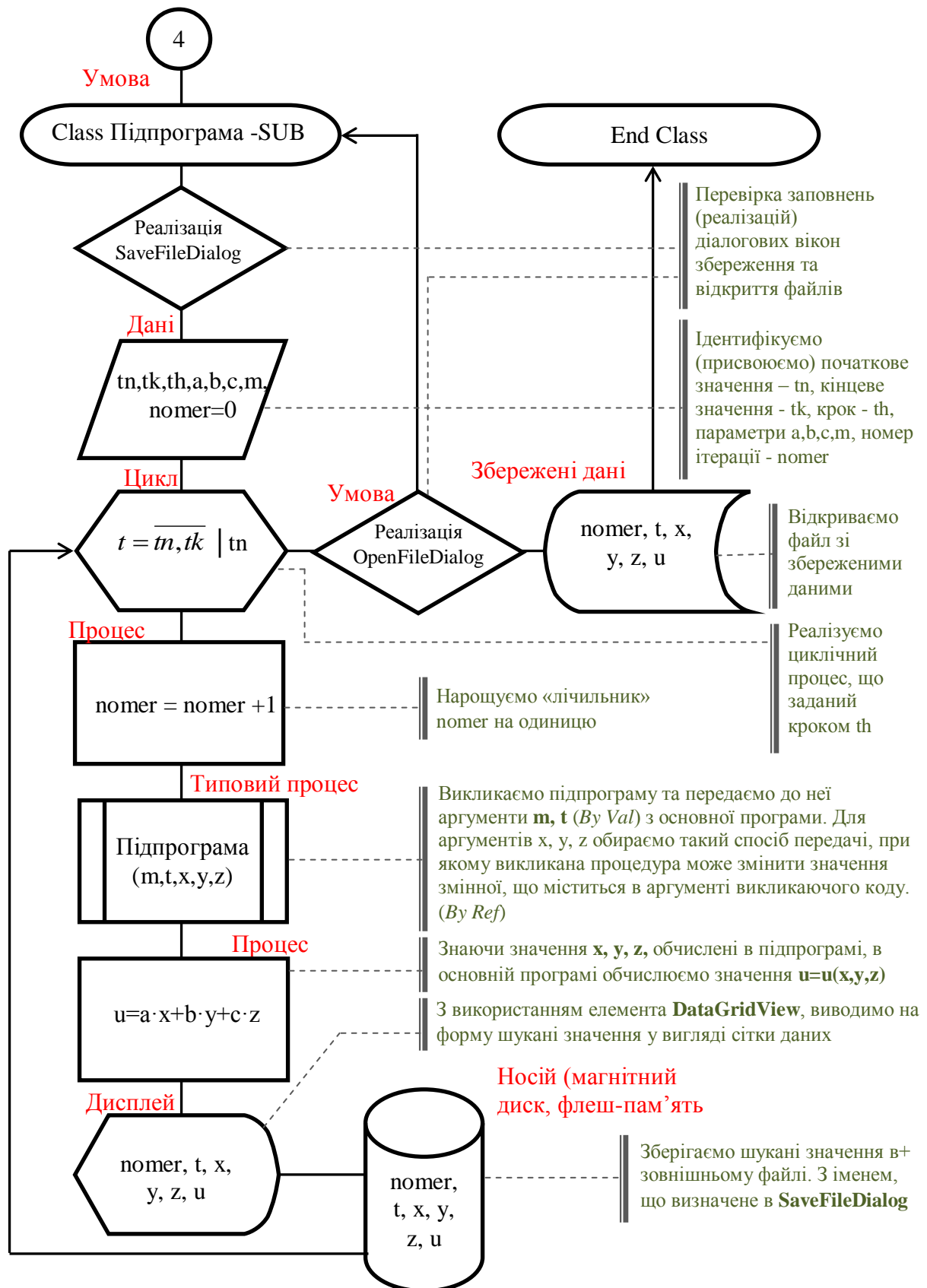


Рисунок 3.10 – Блок-схема основної програми моделі «Процедура – Sub»

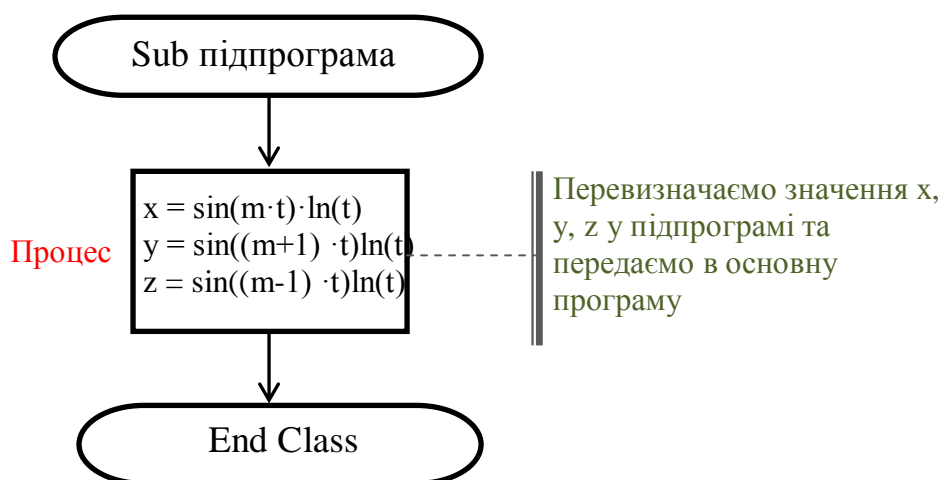


Рисунок 3.11 – Блок-схема підпрограми моделі «Процедура – Sub»

## Лістинг програми моделі «Процедура – Sub»

## 1. Лістинг основного модуля

---

```

Imports System.Math
Public Class Підпрограма_SUB
'Стандартное описание [Dim] всех используемых элементов в последующем коде:
  Dim t As Double, tn As Double, tk As Double, th As Double, a As Double, _
  b As Double, c As Double, m As Double
  Dim x As Double, y As Double, z As Double
  Dim format As New DataGridViewCellStyle
  'DataGridViewCellStyle-Представляет сведения о форматировании и стиле,
  ' применяемые к отдельным ячейкам в элементе управления DataGridView
  ' (ОПИСЫВАЕМ ВО ВНЕШНЕЙ ОБЛАСТИ!).
#Region "ДАННЫЙ УЧАСТОК КОДА ОТВЕЧАЕТ ЗА ФОРМАТИРОВАНИЕ И ВИДИМОСТЬ _
  ИСПОЛЬЗОВАННЫХ ОБЪЕКТОВ"
  Private Sub Підпрограма_SUB_Load(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles MyBase.Load
    With format 'Определяем, как будет определено
      'форматирование ячеек в DataGridView.
      .BackColor = Color.DarkSalmon 'знаем
      .ForeColor = Color.Honeydew 'знаем
    End With : PictureBox2.Visible = False : Label9.Visible = False 'знаем
  End Sub
  Private Sub Button1_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button1.Click
    GroupBox1.Show() : PictureBox2.Visible = True : Label9.Visible = True
    DataGridView1.Columns.Clear() : Label1.Visible = True
  End Sub
#End Region
#Region "НЕПОСРЕДСТВЕННЫЙ КОД ДЛЯ РЕАЛИЗАЦИИ ИСХОДНОЙ МОДЕЛИ, _
  (т.е. суть задачи)."
  Private Sub Вызов_подпрограммы_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Вызов_подпрограммы.Click
    If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
      My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, _
        vbCrLf, False)
    
```

```

tn = CDb1(Val(TextBox1.Text)) 'здесь и далее переопределяем все
    'параметры входящие в модель
tk = CDb1(Val(TextBox2.Text)) ' здесь и далее Val-возвращает числа,
    'содержащиеся в строке в качестве числовых значений
    'соответствующего типа.
th = CDb1(Val(TextBox3.Text))
a = CDb1(Val(TextBox4.Text))
b = CDb1(Val(TextBox5.Text))
c = CDb1(Val(TextBox6.Text))
m = CDb1(Val(TextBox7.Text))
Label1.Visible = False
GroupBox1.Visible = False
Dim шапка() As String = {"№ п.п", "t", "X", "Y", "Z", "U"}
'Задаём в одномерный массив заголовки столбцов таблицы
Dim номер As Integer 'определяем переменную [номер], которую будем
    'использовать как счетчик в автоматическом (циклическом)
    'добавлении строк таблицы - DataGridView1
With DataGridView1 'With- выполняет блок операторов с
    'повторяющимися ссылками на один объект
    'в данном случае [DataGridView1].
    .DefaultCellStyle.BackColor = Color.Bisque
    'DataGridView.DefaultCellStyle - свойство, _
    'возвращает или задает стиль ячейки по умолчанию,
    ' которое будет применяться к ячейкам _
    ' в объекте DataGridView1, если не заданы какие-либо другие
    ' свойства стиля ячейки.
    .GridColor = Color.Red 'Возвращает или задает цвет линий сетки,
    ' разделяющих ячейки.
    For i = 0 To 5
        .Columns.Add(i, шапка(i))
        'где [i]-имя по которому будет упоминаться столбец;
        ' [шапка(i)]- текст для заголовка столбца.
    Next
    For t = tn To tk Step th
        номер = номер + 1
        'наращиваем значение счётчика строки на единицу.
        Call подпрограмма(m, t, x, y, z)
        'вызываем подпрограмму и передаем
        'ей в качестве аргументов переменные: m, t, x, y, z
        Dim u As Double = a * x + b * y + c * z
        'значение функции [u]-вычисляем-
        'в основной программе (см. требование листа задания).
        .Rows.Add(номер, t, x, y, z, u)
        'в DataGridView1, добавляем новую строку,
        'и записываем в неё последовательно значения:
        'номер, t, x, y, z, u
        'Сразу после вывода строки в DataGridView1,
        'записываем её во внешний файл, определенный с использованием
        'диалогового окна [SaveFileDialog1]
My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, _
    номер & t & x & y & z & u, True)
    Next
    .Columns(2).DefaultCellStyle = format
    'форматирование 2-го столбца на основе делегата format
    .Columns(3).DefaultCellStyle = format
    'форматирование 3-го столбца на основе делегата format

```



```

        .Columns(4).DefaultCellStyle = format
        'форматирование 4-го столбца на основе делегата format
    End With
    PictureBox2.Visible = False : Label9.Visible = False 'знаем
End If
End Sub
#End Region
#Region "Данные элементы стандартные для любого диалогового окна _
SaveFileDialog и DataGridView(просто скопируй)"
Private Sub Подпрограмма_SUB_Paint(ByVal sender As Object, _
    ByVal e As System.Windows.Forms.PaintEventArgs) Handles Me.Paint
    With SaveFileDialog1
        ' Добавьте расширение по умолчанию, если пользователь забыл
        ' добавить расширение. По умолчанию: True.
        .AddExtension = True
        ' Если пользователь не указывает расширение и если
        ' свойство AddExtension имеет значение True, используйте это
        ' расширение. По умолчанию: "".
        .DefaultExt = ".txt"
        ' .DefaultExt = ".xls"
        ' По умолчанию: "".
        .Filter = _
        "Text files (*.txt)|*.txt|All files|*.*|Excel files (*.xls)|*.xls"
    End With
    '-----
    With OpenFileDialog1
        ' Как в VB6, используйте набор пар фильтров,
        ' разделенных знаком "|". Каждая
        ' пара включает спецификацию описание|файл.
        ' Используйте "|" между парами. Не указывайте ' "|" в конце.
        .Filter = _
        "Text files (*.txt)|*.txt|All files|*.*|Excel files (*.xls)|*.xls"
    End With
End Sub
#End Region
#Region "Кнопка отвечающая за открытие текстового файла записанного _
Ранее (просто скопируй)"
Private Sub Butt_OPEN_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Butt_OPEN.Click
    If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
        MsgBox(My.Computer.FileSystem.ReadAllText(OpenFileDialog1.FileName))
    End If
End Sub
#End Region
End Class

```

## 2. Лістинг модуля «Процедура – Sub»

---

```

Imports System.Math
Module Module1
    Sub подпрограмма(ByVal m, ByVal t, ByRef x, ByRef y, ByRef z)
        'ByRef-указывает такой способ передачи аргумента, при котором вызванная
        'процедура может изменить значение переменной, содержащейся в аргументе
        'вызывающего кода.
        'ByVal -указывает такой способ передачи аргумента, при котором вызванная

```

```

'процедура НЕ может изменить значение переменной, содержащейся в аргументе
'вызывающего кода.
  x = Sin(m * t) + Log(Abs(t))
  y = Sin((m + 1) * t) + Log(Abs(t))
  z = Sin((m - 1) * t) + Log(Abs(t))
End Sub
#Region "ЗАМЕЧАНИЯ: Изменение значения аргумента процедуры"
'При вызове процедуры каждый аргумент соответствует одному из параметров,
'определенных в процедуре.
'В некоторых случаях код процедуры может изменять значение основного
'аргумента в вызывающем коде. В других случаях процедура может изменять
'только его локальную копию.

'При вызове процедуры Visual Basic создает локальную копию каждого
'аргумента, который передается ByVal
'Для каждого передаваемого аргумента ByVal (Visual Basic) Visual Basic
'предоставляет для кода процедуры прямую ссылку на элемент программирования,
'содержащегося в аргументе в вызывающем коде.

'Если базовый элемент вызывающего кода является изменяемым, и аргумент
'передается ByVal, то код процедуры позволяет изменить значение элемента
'в коде вызова при помощи прямой ссылки.
#End Region
End Module

```

---

### 3.6 Реалізація моделі «Процедура – Function»

Умова. Керівництво інвестиційної компанії «Millennium Finance», в рамках програми освоєння нових ринків, вирішило розглянути рівень прибутковості інвестицій при придбанні пакету акцій логістичних компаній. На основі зібраної статистичної інформації було побудовано аналітичну модель залежності рівня прибутковості акцій ( $y$ ) на ринку логістичних послуг у залежності від темпів зростання капіталізації компаній ( $x$ ):

$$y = \sqrt{\sin(1 + 2x^3)}.$$

У якості компаній, що потенційно можуть мати високий рівень прибутковості, було обрано 4 достатньо молодих, але перспективних логістичних компаній та отримано дані про динаміку темпів зростання їх капіталізації ( $x$ ):  $\bar{x}_i = \{0,2; 1,2; -0,2; 4,5\}$ .

У допомогу аналітику необхідно розробити біржевий калькулятор реалізації розрахунків зі зручним інтерфейсом (рисунки 3.12 – 3.13).

При цьому: а) визначення функції рівня прибутковості акцій ( $y$ ) виконати за допомогою процедури-функції; б) значення динаміки темпів зростання капіталізації ( $x$ ) завантажити із файла; в) отримані результати розрахунків зберегти в зовнішній файл, для використання в процесі «навчання» побудованої моделі шляхом обробки цих даних за допомогою відповідного програмного забезпечення та візуалізації результатів.

Блок-схему алгоритму подано на рисунках 3.14–3.15.

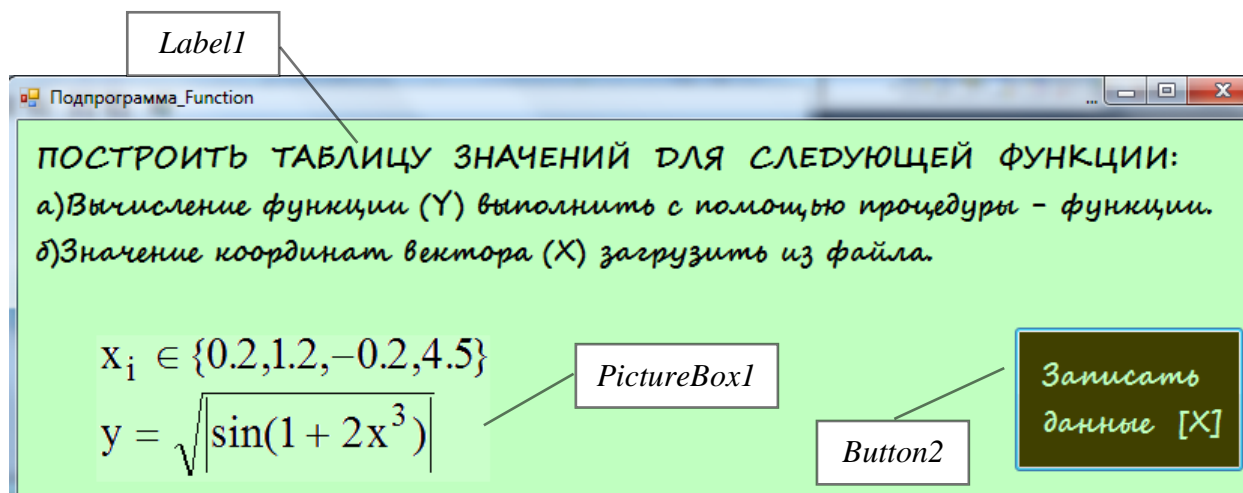


Рисунок 3.12 – Графічний інтерфейс користувача моделі «Підпрограма –Function»



Рисунок 3.13 – Графічний інтерфейс користувача моделі «Підпрограма –Function»

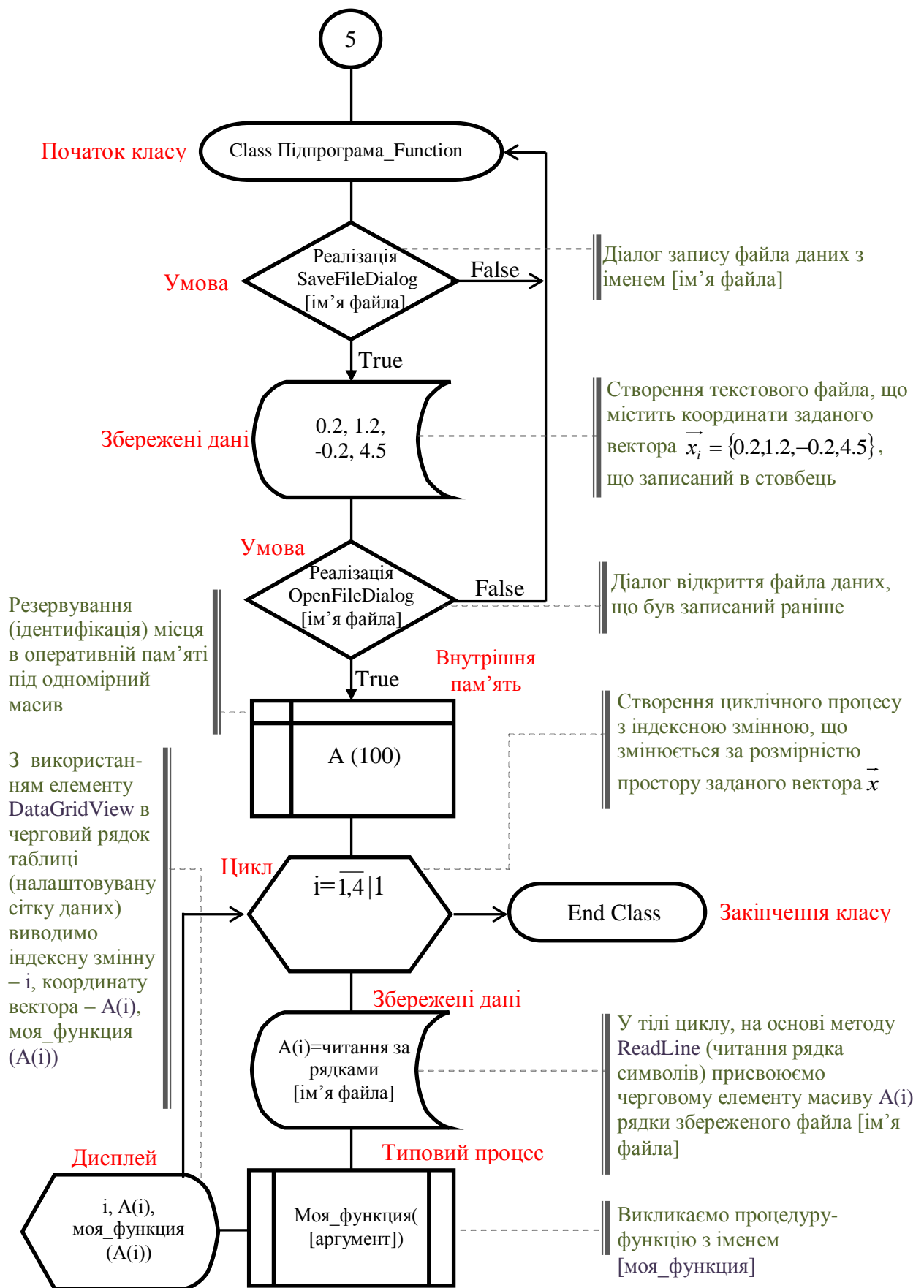


Рисунок 3.14 – Блок-схема основної програми моделі «Підпрограма –Function»

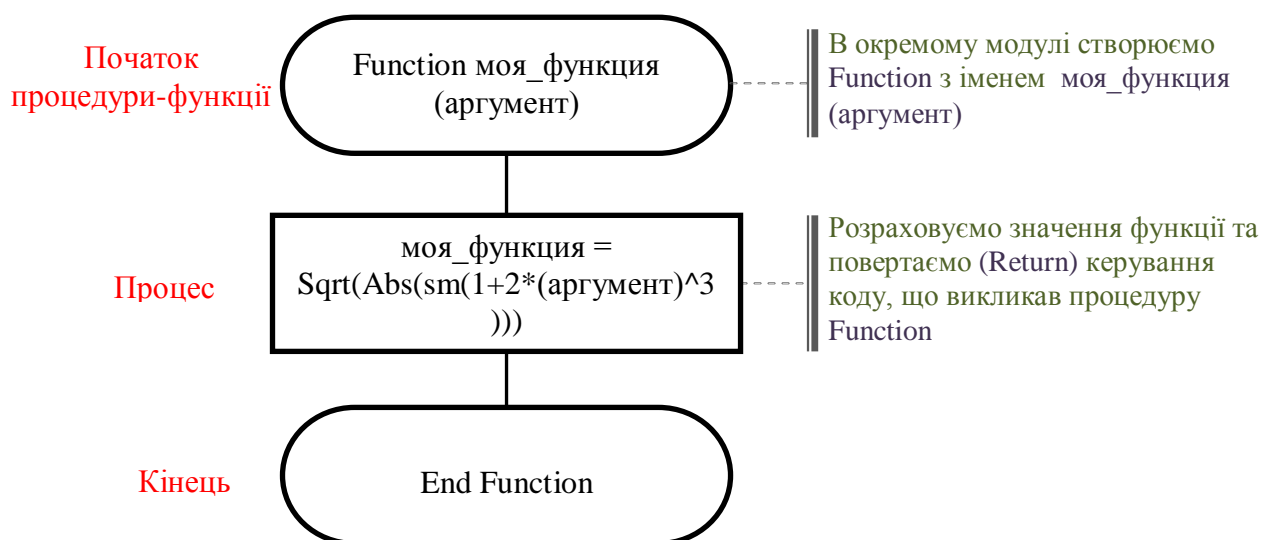


Рисунок 3.15 – Блок-схема підпрограми моделі «Підпрограма – Function»

## Лістинг програми моделі «Процедура – Function»

## 1. Лістинг основного модуля

```

Public Class Подпрограмма_Function
'Это имя формы, где выполняется 5-я задача
#Region "Кнопка, отвечающая за открытие и считывание построчно символов _
текстового файла записанного ранее"
Private Sub Button1_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button1.Click
DataGridView1.Rows.Clear() : DataGridView1.Visible = True
'Программируем диалоговое окно открытия файла и считываем
'указанный файл в массив A(i)

If OpenFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
Dim A(100) As Single, i As Integer
Dim Читатель As New IO.StreamReader(OpenFileDialog1.FileName)
'где: Пространство имен System.IO-
'содержит типы, позволяющие осуществлять чтение и запись в файлы
'StreamReader-реализует TextReader, который считывает символы из
'потока байтов в определенной кодировке.
'где TextReader-представляет средство чтения, позволяющее считывать
'последовательные наборы знаков.
For i = 1 To 4
A(i) = CDb1(Читатель.ReadLine)
'ReadLine - метод выполняет чтение строки символов из текущего
'потока и возвращает данные в виде строки.
'В этой точке программного кода сформирован одномерный массив A(i),
'приступаем к обработке процедуры типа FUNCTION,
'для этого добавь Module2.vb и сформируй в нём функцию
'с листа задания со следующим синтаксисом записи:
'Function имя_функции[(список_параметров)] As тип_возвращаемого_значения
'MsgBox(i & "-->" & Sin(A(i))) 'Проверь, как читает выбранный файл
'MsgBox(моя_функция(моя_функция(A(i))))
'Проверь, как работает [моя_функция]
'_____ ДАЛЬНЕЙШИЙ КОД ЗАНЕСЕТ ПОЛУЧЕННЫЕ ДАННЫЕ В ТАБЛИЦУ НА ФОРМЕ _____
DataGridView1.Rows.Add(i, A(i), моя_функция(A(i)))
  
```

```

        'Всё, данные уже на таблице
    Next
    Читатель.Close()
    'Закрывает объект и освобождает все системные ресурсы,
    'связанные с устройством чтения.
End If
End Sub
#End Region
#Region "Записываем данные из текстового поля в текстовый файл _
на основе диалога - SaveFileDialog1"
Private Sub Данные_для_X_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Данные_для_X.Click
    TextBox1.Visible = True : Button1.Visible = True
    'Данный диалог сохранения файла был описан ранее
    If SaveFileDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then
        My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, _
            TextBox1.Text, False)
    End If
End Sub
#End Region
End Class

```

## 2. Лістинг модуля «Процедура – Function»

---

```

Imports System.Math
Module Module2
    Function моя_функция(ByVal аргумент) As Single
        'Return аргумент + 1 'Сначала проверь простое выражение для контроля,_
        'потом записывай функцию из листа задания:
        Return Sqrt(Abs(Sin(1 + 2 * (аргумент) ^ 3)))
        'Return-Возвращает управление коду, вызвавшему процедуру Function
    End Function
End Module

```

---

## 4 ГРАФІЧНІ МОЖЛИВОСТІ МОВИ VB

### 4.1 Загальні відомості про векторну графіку

Програмна побудова ліній, прямокутників та інших фігур відбувається в системі координат. За замовчуванням використовується плоска декартова система координат, початок координат якої розташовано у верхньому лівому куті екрана, вісь  $X$  спрямовано вправо, а вісь  $Y$  – униз. Одиницею виміру в заданій за замовчуванням системі координат є точка / піксель (мінімальний елемент зображення) (рисунок 4.1).

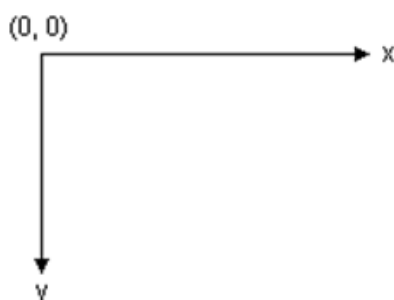


Рисунок 4.1 – Вигляд системи координат на моніторі

Зображення на моніторі комп'ютера формується як прямокутний масив точок, що є мінімальними елементами зображення. Кількість точок, які відображаються на екрані, залежить від типу монітора та може бути змінена користувачем у певних межах (рисунок 4.2).

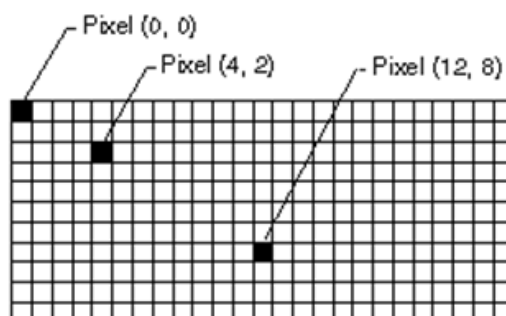


Рисунок 4.2 – Принцип формування зображення на моніторі

За час розвитку комп'ютерної графіки було виділено базові геометричні фігури (примітиви), що найбільш корисні при створенні двомірних зображень:

1. Лінії – Lines.
2. Прямокутники – Rectangles.
3. Еліпси – Ellipses.
4. Дуги – Arcs.
5. Багатокутники – Polygons.
6. Фундаментальні сплайни (криві) – Splines (Curves).

## 7. Сплайни (криві) Безьє – Bezier Splines (Curves).

Для роботи з графічними можливостями платформи .Net необхідно володіти основними програмними інструкціями (таблиця 4.1).

Таблиця 4.1 – Інструкції для роботи з графікою платформи .Net

Інструкція 1	Опис 2
ArrowAnchor	Задає маркер у формі стрілки
Brushes	Пензлики для кожного зі стандартних кольорів
Control.CreateGraphics	Задає об'єкт Graphics для елемента керування
Clear(Color.White)	Очищає всю поверхню рисування й виконує заливання поверхні зазначеним кольором фону
Dash	Задає лінію, що складається зі штрихів
DashStyle	Задає стиль пунктирних ліній, що нарисовані за допомогою об'єкта Pen
DiagonalCross	Шаблон для перехресних діагональних ліній
DrawClosedCurve()	Будує замкнену фундаментальну криву, що задана масивом структур Point
DrawCurve	Будує фундаментальну криву через точки зазначеного масиву структур Point
DrawPolygon	Рисую багатокутник, що заданий масивом структур Point
DrawString	Створює текстовий рядок, який вказується в заданому місці за допомогою заданих об'єктів Brush і Font
EndCap	Отримує або задає стиль закінчення ліній, що нарисовані за допомогою об'єкта Pen
FillClosedCurve	Заповнює внутрішню частину замкненої фундаментальної кривої, що задана масивом структур Point
FromArgb(a,r,g,b)	Створює структуру Color із зазначених 8-розрядних значень компонентів ARGB (альфа, червоний, зелений і синій)
Graphics.DrawLine	Проводить лінію, що з'єднує дві точки, що задаються парами координат
Graphics.RotateTransform	Застосовує задане обертання до матриці перетворення даного об'єкта Graphics
Graphics.TranslateTransform	Змінює початок координат координатної системи шляхом додавання заданого зміщення до матриці перетворення даного об'єкта Graphics
HatchBrush	Задає прямокутний пензлик зі стилем штрихування, основним кольором і кольором фону
HatchStyle	Указує різні шаблони, що доступні для об'єктів HatchBrush
LineCap	Задає доступні стилі відрізка, за допомогою яких об'єкт Pen завершує рядок
Pen	Визначає об'єкт, що використовується для рисування прямих ліній і кривих
PointF	Представляє впорядковану пару координат X та Y із плаваючою комою, що визначає точку на двовимірній площині
Refresh()	Примусово створює умови, при яких елемент керування робить недоступною свою клієнтську область і негайно перерисовує себе й усі дочірні елементи
SolidBrush	Визначає пензлик одного кольору. Пензлики використовуються для заливання графічних фігур, таких як прямокутники, еліпси, кола, багатокутники й контури
Drawing.Drawing2D	Простір імен System.Drawing.Drawing2D надає розширені функціональні можливості векторної й двомірної графіки



Покажемо на конкретних прикладах ці можливості.

Щоб побудувати лінію, потрібно створити два об'єкти: об'єкт `Graphics` і об'єкт `Pen`. Об'єкт `Graphics` містить методи, що безпосередньо виконують побудову графічних об'єктів, а об'єкт `Pen` служить сховищем атрибутів, таких як колір, ширина, стиль лінії тощо.

### Побудова лінії

Щоб побудувати лінію (рисунок 4.3), викличте метод `DrawLine` об'єкта `Graphics`. Об'єкт `Pen` передається методу `DrawLine` у якості одного з аргументів. У наведеному нижче прикладі демонструється побудова відрізка, що з'єднує точки з координатами  $(0, 0)$  і  $(100, 100)$ :

---

```
Public Class Form1
    Private Sub PictureBox1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles PictureBox1.Click
        Dim graphics As Graphics 'идентифіцируем поверхность рисования
        graphics = PictureBox1.CreateGraphics 'PictureBox1 определяем как
        'поверхность рисования (дальнейшие построения производятся на PictureBox1)
        Dim myPen As New Pen(Brushes.Brown, 6) 'определяем кисть _
        'коричневого цвета, толщиной 6 условных единиц
        graphics.DrawLine(myPen, 0, 0, 100, 100) ' на поверхности рисования _
        'строим линию кистью myPen, с началом в точке (0,0) и концом (100,100)
    End Sub
End Class
```

---



Рисунок 4.3 – Формування зображення лінії

### Штрихові лінії й завершення відрізків

Об'єкт `Pen` також дозволяє змінювати значення деяких своїх властивостей, таких як `DashStyle`, що дозволяє задавати характеристики лінії.

Змінюючи значення властивостей об'єкта `Pen`, можна задати багато атрибутів лінії. Властивості `StartCap` і `EndCap` визначають вид кінцевих точок відрізка. Кінець відрізка може бути плоским, квадратним, круглим, трикутним або мати довільну форму. На наведеному нижче прикладі демонструються різні стилі завершення та з'єднання ліній (рисунок 4.4).



Рисунок 4.4 – Приклад стилів кінців відрізка

У наведеному нижче прикладі демонструється побудова штрихових ліній і завершення відрізків (рисунок 4.5).

```
Public Class Form1
    Private Sub PictureBox1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles PictureBox1.Click
        Dim graphics As Graphics 'ідентифіцируем поверхню рисования
        graphics = PictureBox1.CreateGraphics 'PictureBox1 определяем как
        'поверхность рисования (дальнейшие построения производятся на
        PictureBox1)
        Dim myPen As New Pen(Brushes.Brown, 6) 'определяем кисть
        'коричневого цвета, толщиной 6 условных единиц
        graphics.DrawLine(myPen, 10, 10, 100, 100) ' на поверхности рисования
        'строим линию кистью myPen, с началом в точке (0,0) и концом (100,100)
        myPen.DashStyle = Drawing2D.DashStyle.Dot 'задаём стиль пунктирных линий
        myPen.StartCap = Drawing2D.LineCap.RoundAnchor 'задаём стиль начала отрезка
        myPen.EndCap = Drawing2D.LineCap.ArrowAnchor 'задаём стиль конца отрезка
        graphics.DrawLine(myPen, 100, 100, 200, 50) 'продолжаем рисовать
        'штрих-пунктирную линию
    End Sub
End Class
```

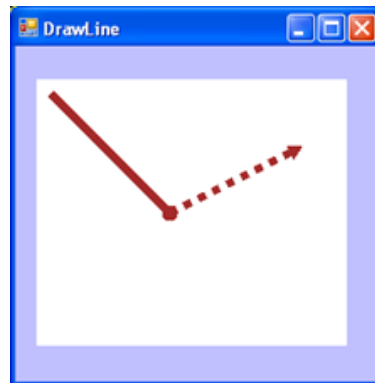


Рисунок 4.5 – Приклад стилів ліній та кінців відрізка

### Побудова прямокутника

Побудова прямокутників (рисунок 4.6) схожа на побудову ліній. Щоб побудувати прямокутник, потрібно створити два об'єкти: об'єкт `Graphics` і об'єкт `Pen`. В об'єкта `Graphics` є метод `DrawRectangle`. Об'єкт `Pen` передається методу `DrawRectangle` у якості одного з аргументів. У наведеному нижче прикладі демонструється побудова прямокутника шириною 150 і висотою 100 точок, верхній лівий кут якого розташований у точці з координатами (10, 10):

---

```

Public Class Form1
    Private Sub PictureBox1_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles PictureBox1.Click
        Dim graphics As Graphics 'идентифицируем поверхность рисования
        graphics = PictureBox1.CreateGraphics 'PictureBox1 определяем как
        'поверхность рисования (дальнейшие построения производятся на PictureBox1)
        Dim myPen As New Pen(Brushes.Brown, 6) 'определяем кисть
        'коричневого цвета, толщиной 6 условных единиц
        graphics.DrawRectangle(myPen, 10, 10, 150, 100) ' рисуем прямоугольник,
        'который определен парой координат, шириной и высотой
    End Sub
End Class

```

---

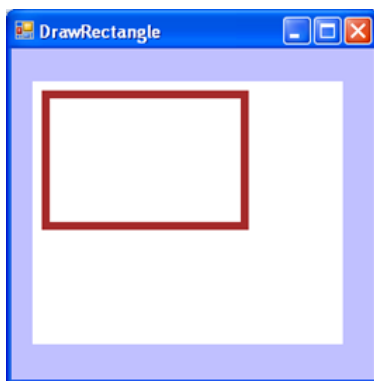


Рисунок 4.6 – Приклад виведення прямокутника на монітор

### Побудова еліпса

Побудова еліпса (рисунок 4.7) відбувається аналогічно іншим графічним примітивам. В об'єкта Graphics є метод DrawEllipse. Об'єкт Pen передається методу DrawEllipse у якості одного з аргументів. Інші аргументи, які передані методу DrawEllipse, задають прямокутник, що обмежує цей еліпс:

---

```

Public Class Form1
    Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox1.Click
        Dim graphics As Graphics 'идентифицируем поверхность рисования
        graphics = PictureBox1.CreateGraphics 'PictureBox1 определяем как
        'поверхность рисования (дальнейшие построения производятся на PictureBox1)
        Dim myPen As New Pen(Brushes.Brown, 6) 'определяем кисть
        'коричневого цвета, толщиной 6 условных единиц
        graphics.DrawRectangle(myPen, 10, 10, 150, 100) ' рисуем прямоугольник,
        'который определен парой координат, шириной и высотой
        graphics.DrawEllipse(myPen, 10, 10, 150, 100) 'рисует эллипс, определяемый
        'ограничивающим прямоугольником, заданным с помощью координат
        'для левого верхнего угла прямоугольника, высоты и ширины
    End Sub
End Class

```

---

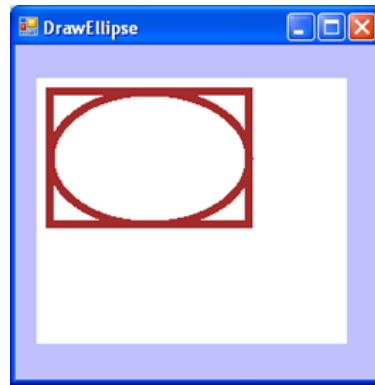


Рисунок 4.7 – Приклад виведення еліпса на монітор

## 4.2 Реалізація моделі «Графіка»

Умова. Реалізувати програмний додаток у вигляді (рисунок 4.8), що використовується для побудови графіків функцій:

$y = -50 \cdot \sin\left(\frac{1}{20} \cdot x\right)$  – функція задана в декартовій системі координат;

$\rho = a \cdot \sin(n \cdot \varphi)$  – функція задана в полярній системі координат,

де  $a$  – радіус пелюстка;

$n$  – кількість пелюстків;

$\rho$  – полярний радіус.

Для перетворення полярних координат у декартові використовується співвідношення:

$$\begin{cases} x = \rho \cdot \cos \varphi \\ y = \rho \cdot \sin \varphi \end{cases}$$

Задачі на побудову графіків функцій мають велике практичне значення з точки зору візуалізації, аналізу та навіть прогнозування розвитку організаційних систем та процесів.

Використовуючи розроблений програмний додаток (рисунок 4.8 та 4.9 а – 4.9 д), аналітик має можливість динамічно змінювати параметри функцій та в реальному часі отримувати результат цих змін у вигляді відповідних змін у графіку. Наприклад, це може бути модель дослідження залежності витрат палива автомобіля при зміні швидкості руху, повної маси транспортного засобу, характеру дорожнього покриття тощо.

Таким чином, по суті, ми отримуємо в своє розпорядження зручний інструмент для моделювання процесів різної природи.

Блок-схеми алгоритму подано на рисунках 4.10–4.12.

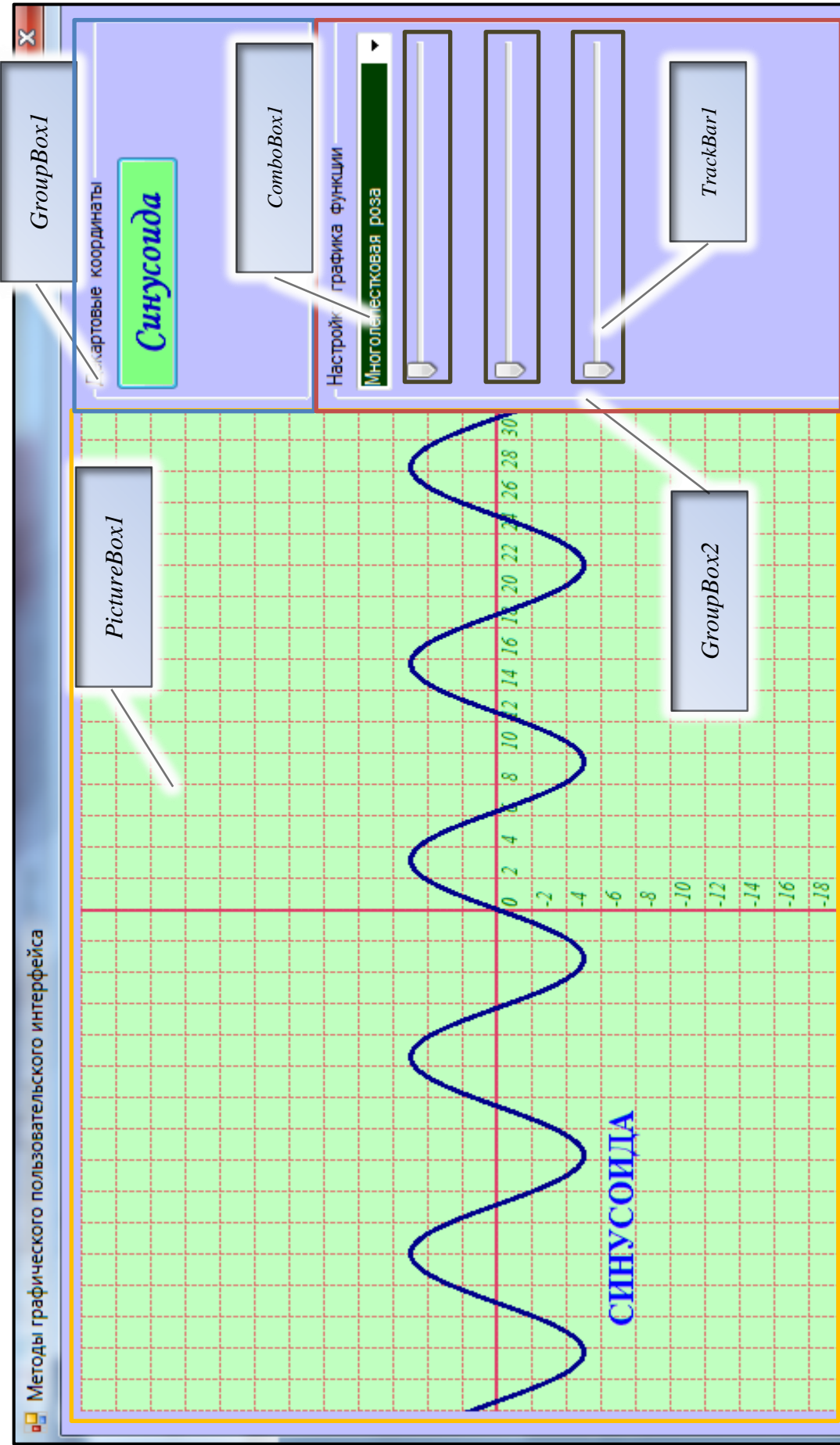


Рисунок 4.8 – Графічний інтерфейс користувача моделі «Графіка»

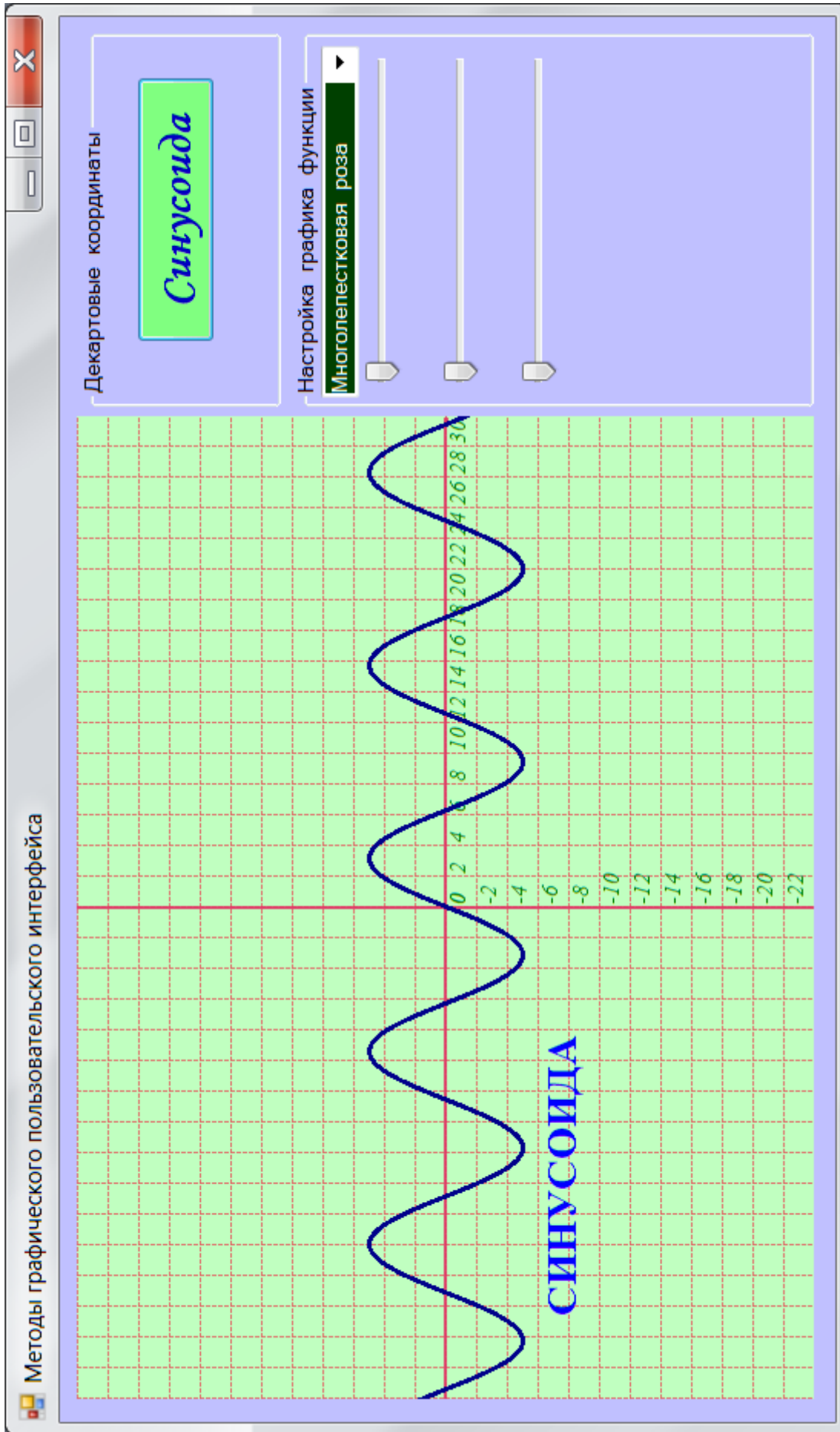


Рисунок 4.9 а – Варианти побудови графіків функцій моделі «Графіка» (синусоида)

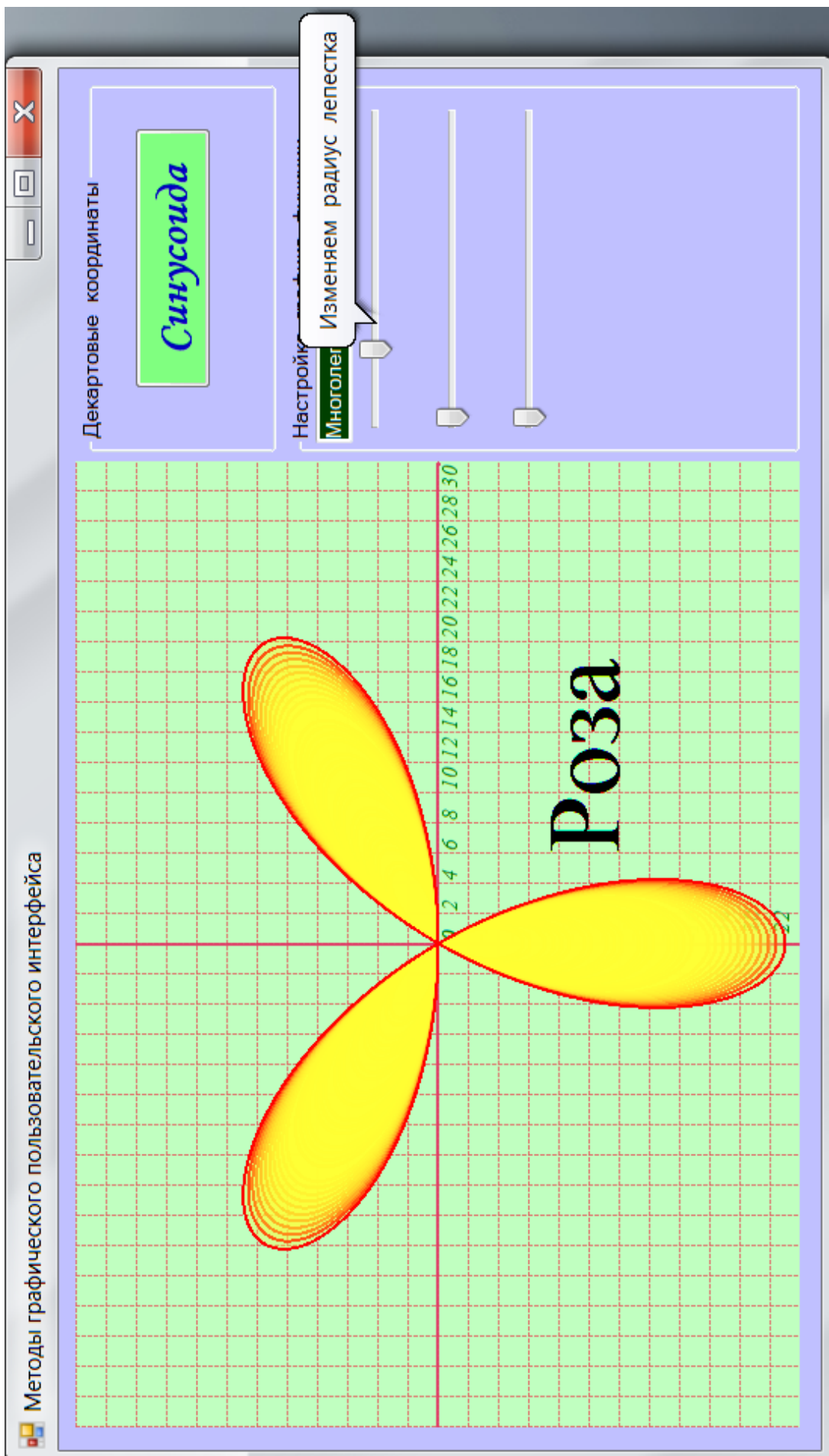


Рисунок 4.9 б – Варіації побудови графіків функцій моделі «Графіка» (зміна радіуса пелюстка графіка «Роза»)

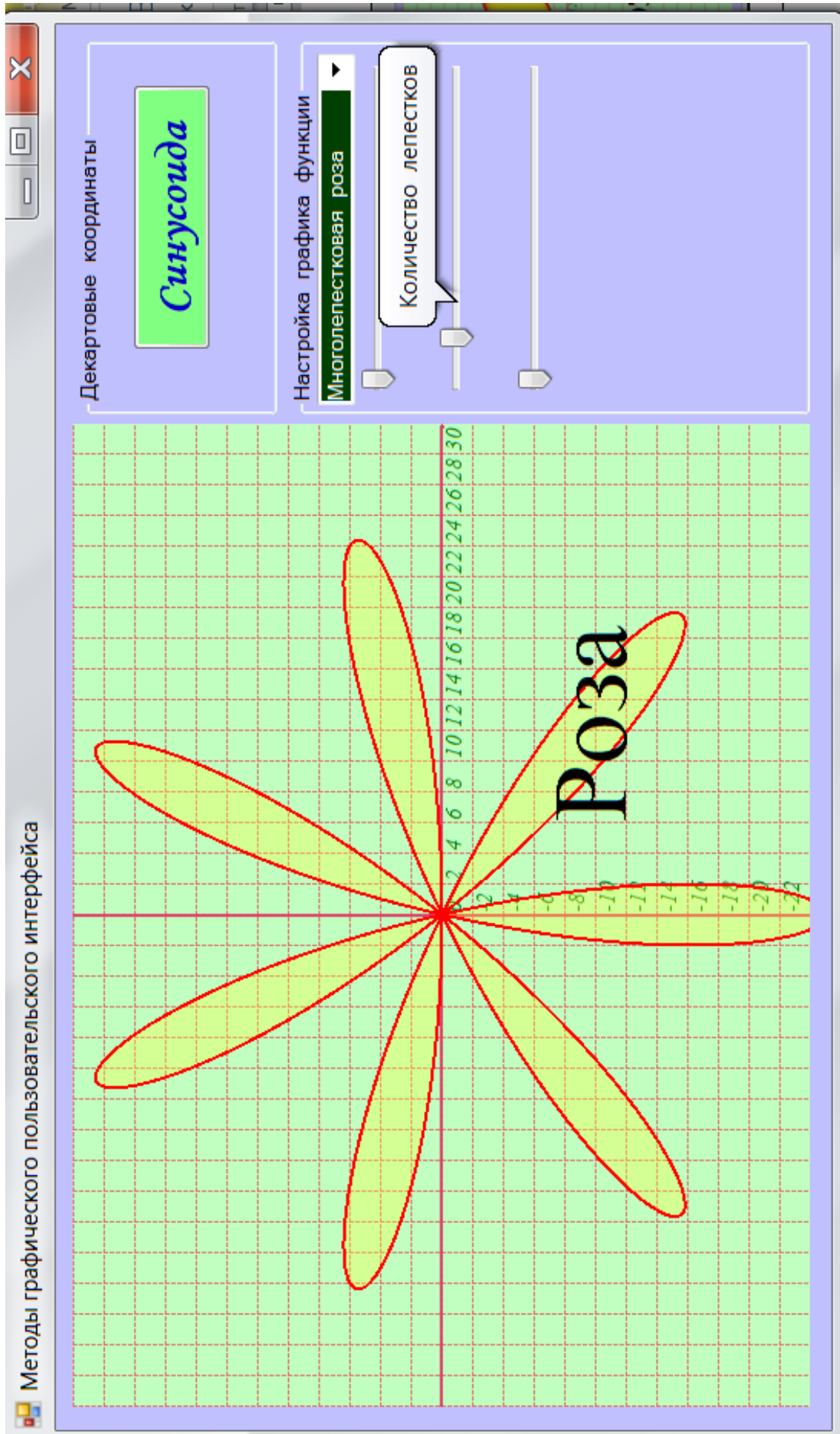


Рисунок 4.9 в – Вариації побудови графіків функцій моделі «Графіка» (зміна кількості пелюсток графіка «Роза»)



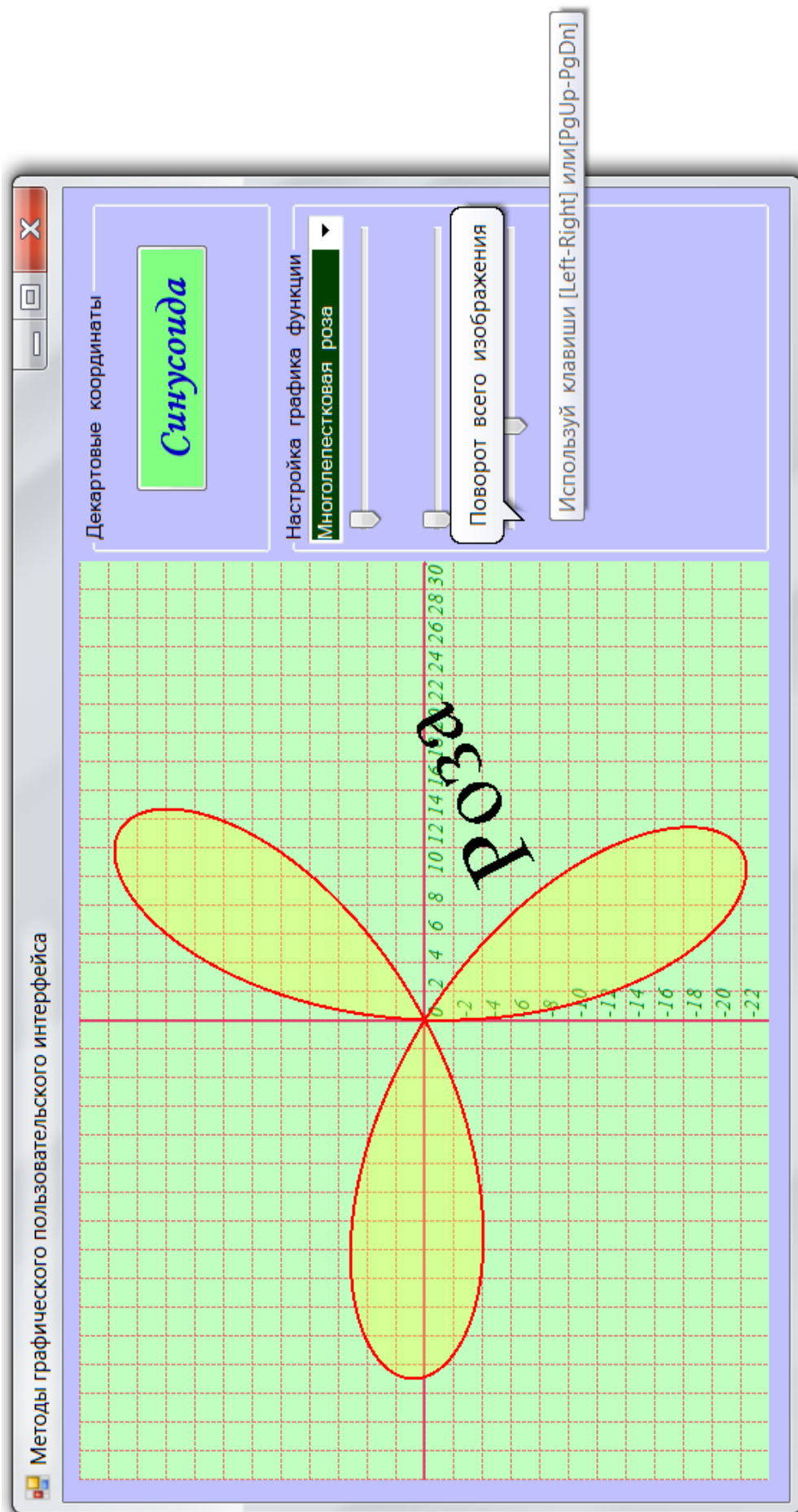


Рисунок 4.9 г – Вариації побудови графіків функцій моделі «Графіка» (зміна кута нахилу графіка «Роза»)

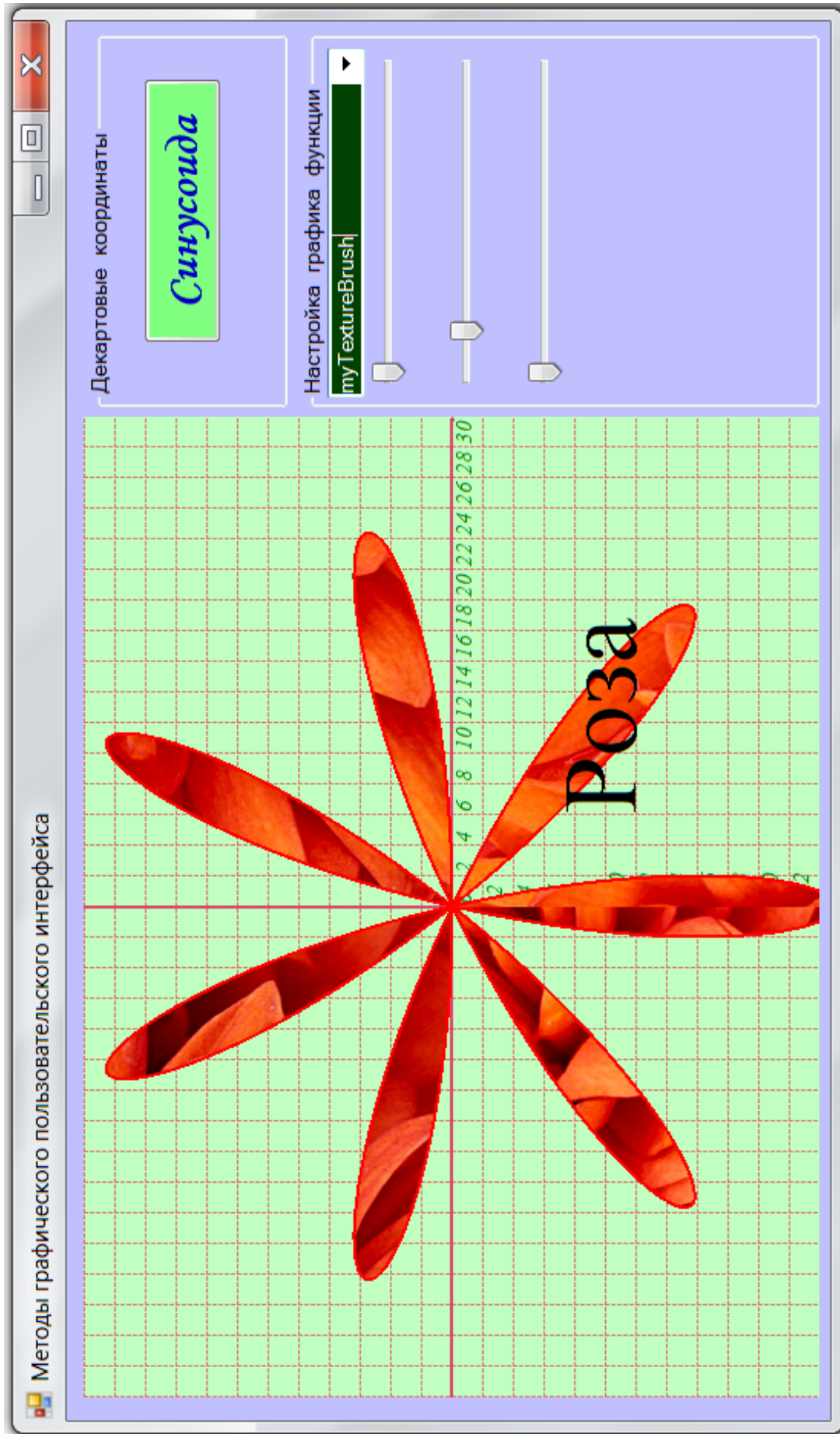


Рисунок 4.9 д – Варианти побудови графіків функцій моделі «Графіка» (заливання графіка «Роза» текстурою)

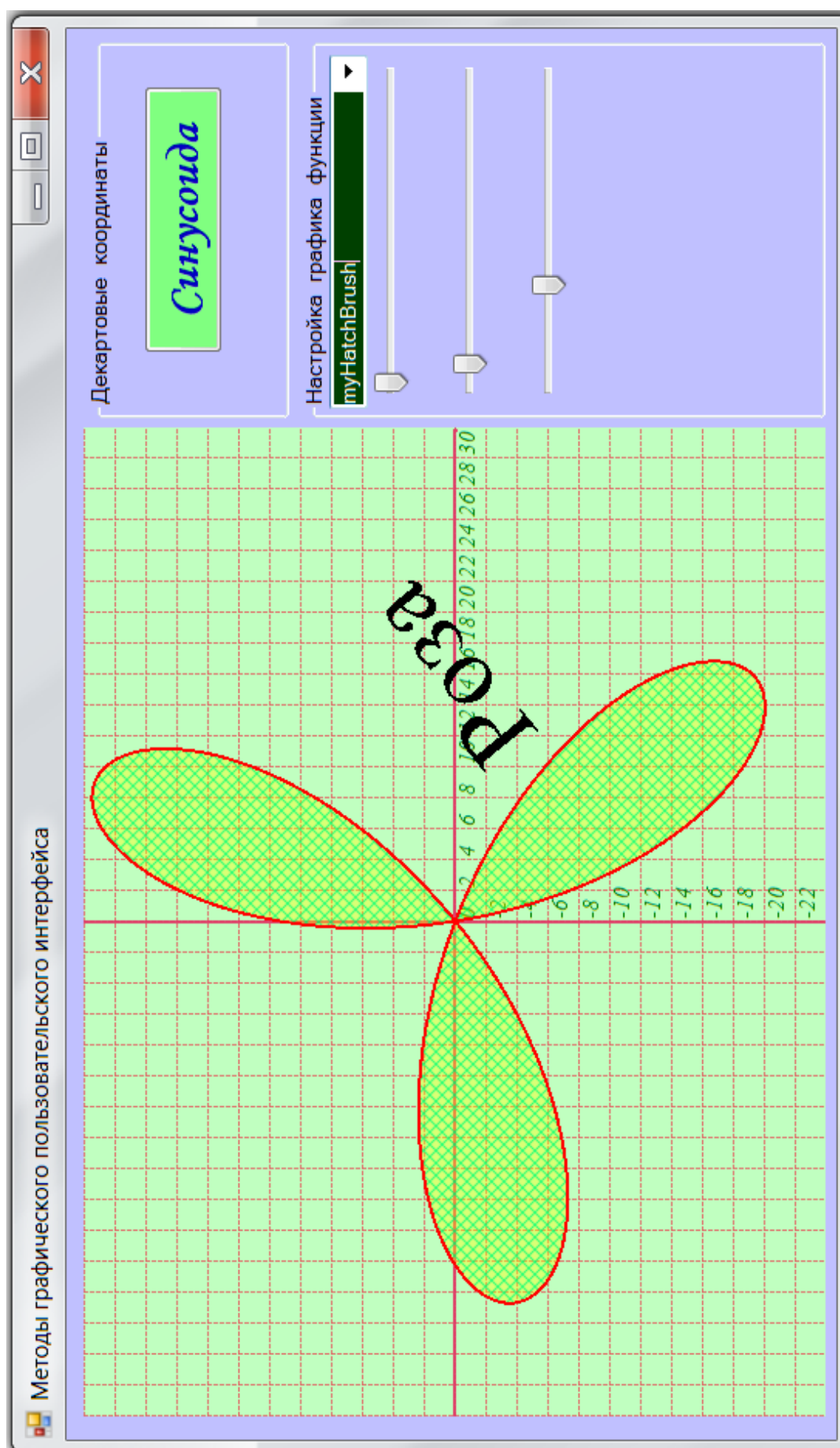


Рисунок 4.9 е – Вариация побудови графіків функцій моделі «Графіка» (заливання графіка «Роза» узором)

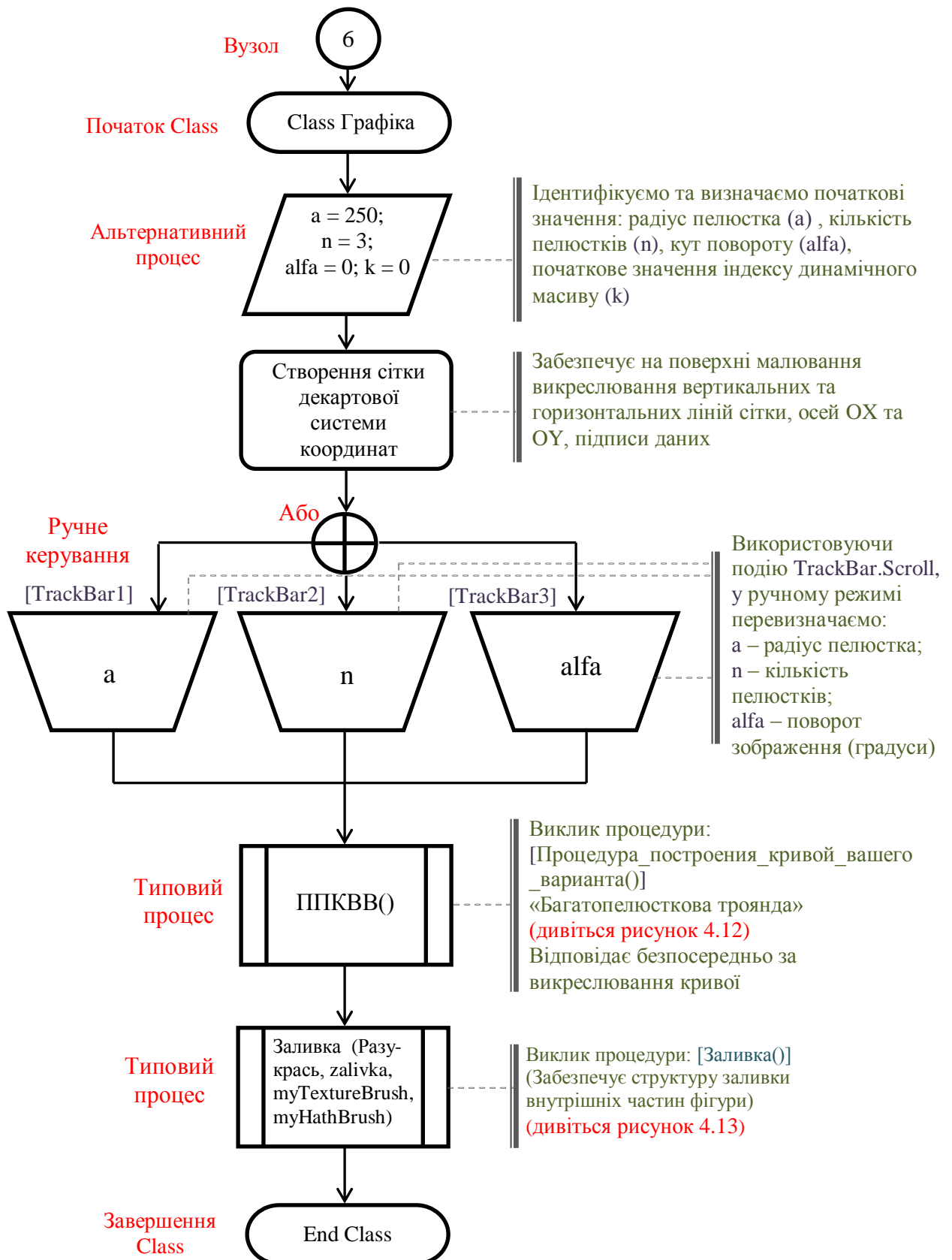


Рисунок 4.10 – Блок-схема основної програми моделі «Графіка»

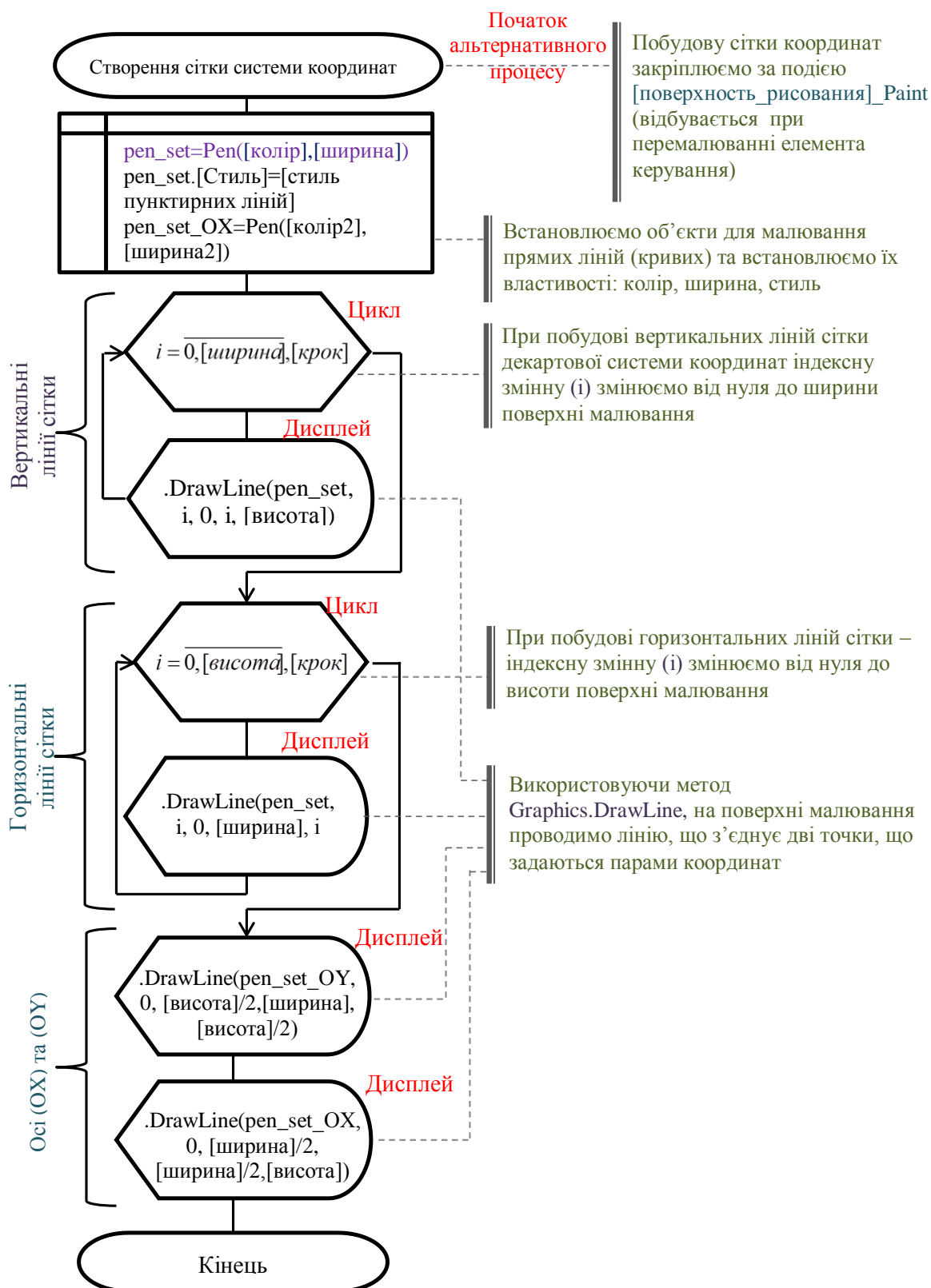


Рисунок 4.11 – Блок-схема альтернативного процесу «Створення сітки для декартової системи координат» моделі «Графіка»

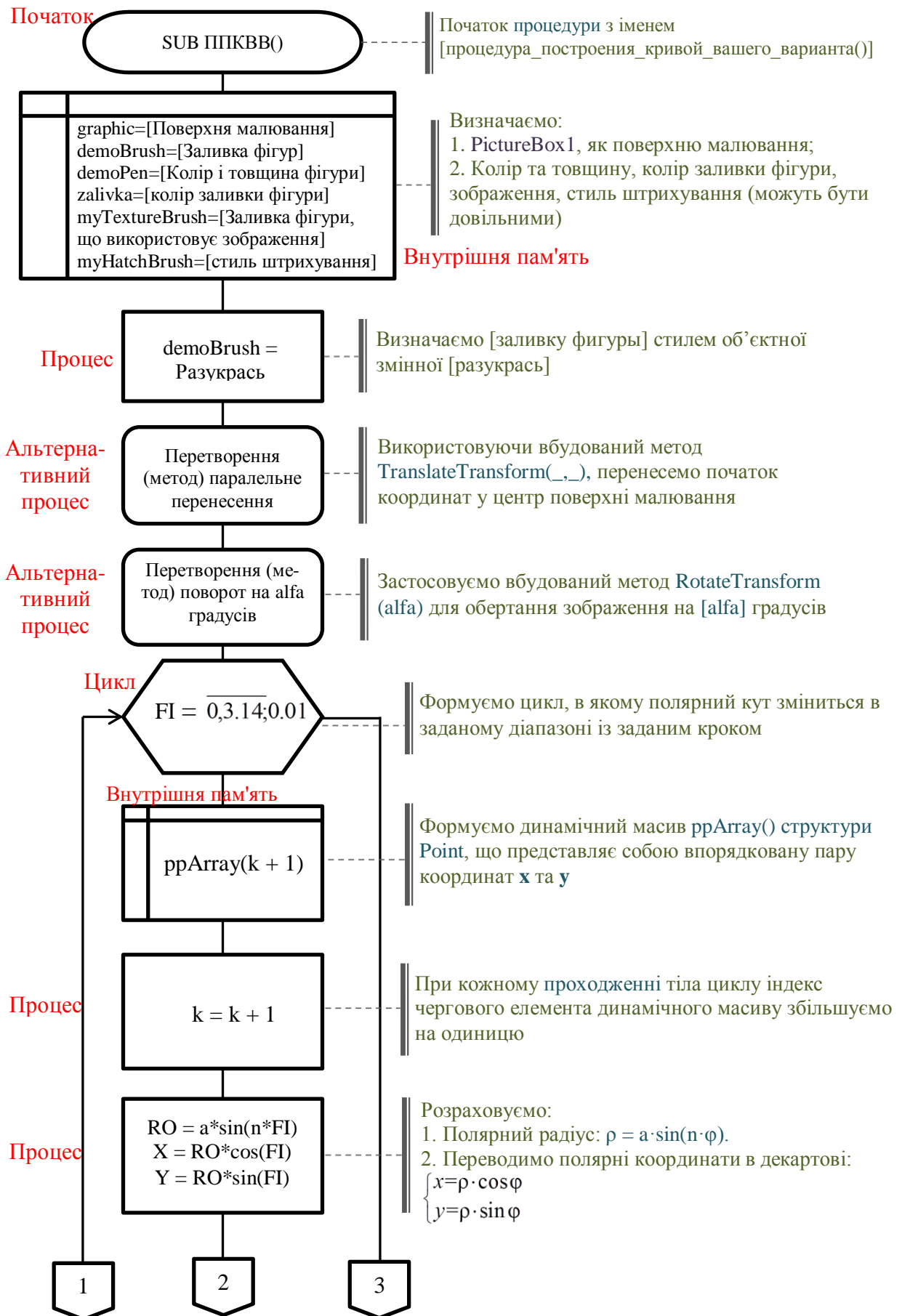


Рисунок 4.12 – Блок-схема процедури побудови кривої «багатопелюсткова роза» моделі «Графіка»

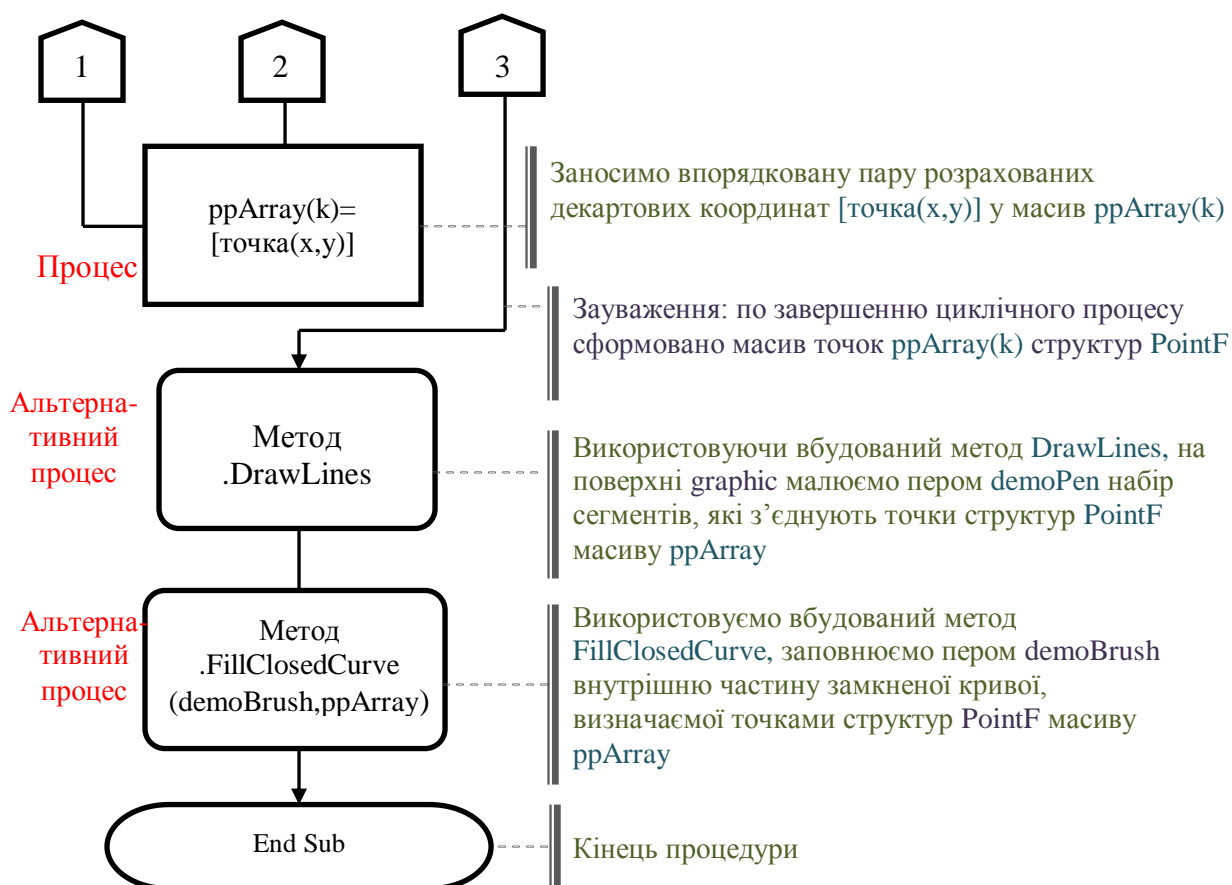


Рисунок 4.12 – Аркуш 2

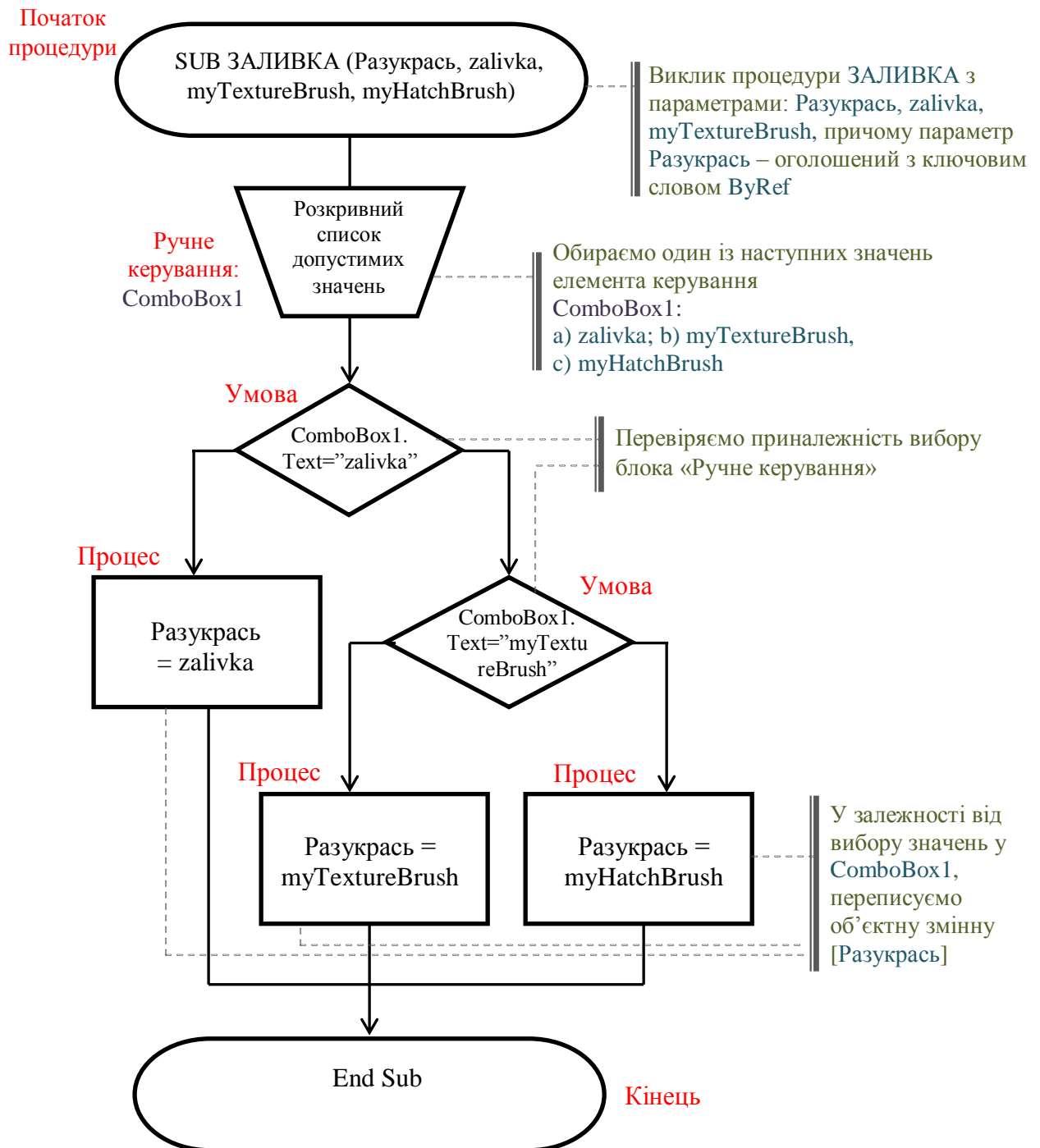


Рисунок 4.13 – Блок-схема процедури «Заливка» моделі «Графіка»

## Лістинг програми моделі «Графіка»

```

Imports System.Math ' это уже известно
Imports System.Drawing.Drawing2D 'Drawing2D<Ru.Рисунок2D>
    'Необходим для работы HatchBrush (т.е. стилем штриховки)
    '<Если забыл, VS добавит>
Public Class Графика
    Public a = 250, n = 3, alfa = 0 'Определяем параметры:
        'длину лепестка, число лепестков, поворот изображения
#Region "Создание сетки декартовой системы координат"
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, _

```



```

ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
    Me.Size = New Size(931, 543)
    PictureBox1.Refresh() ' Refresh<Обновление>-
'Принудительно создает условия, при которых элемент управления делает
'недоступной свою клиентскую область и немедленно перерисовывает себя
'и все дочерние элементы.
    Процедура_построения_кривой_вашего_варианта()
End Sub
Sub PictureBox1_Click1(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles PictureBox1.Click
    PictureBox1.CreateGraphics.Clear (Color.White)
'Сброс графического изображения на PictureBox1
End Sub
Private Sub PictureBox1_Paint(ByVal sender As Object, _
ByVal e As System.Windows.Forms.PaintEventArgs) Handles PictureBox1.Paint
'<Paint-Ru.Краска>
'Paint-Событие-Происходит при перерисовке элемента управления.
    Dim i As Integer
    Dim pen_set As New Pen(Color.FromArgb(225, 225, 100, 100), 0.5)
    pen_set.DashStyle = Drawing2D.DashStyle.Dash
'_____ Вертикальные линии сетки _____
    For i = 0 To PictureBox1.Width Step 20 ' Вертикальные линии сетки
        e.Graphics.DrawLine(pen_set, i, 0, i, CInt(PictureBox1.Height))
        e.Graphics.DrawString(CStr(i) / 10, New Font("Times New Roman", 10, _
        FontStyle.Italic), Brushes.Green, _
        New PointF(i + PictureBox1.Width / 2, PictureBox1.Height / 2))
    Next
'_____ Горизонтальные линии сетки _____
    For i = 0 To PictureBox1.Height Step 20
        e.Graphics.DrawLine(pen_set, 0, i, CInt(PictureBox1.Width), i)
    Next
'_____ Ось OX _____
    Dim pen_set_ox As New Pen(Color.FromArgb(225, 225, 50, 100), 2)
    pen_set_ox.EndCap = Drawing2D.LineCap.ArrowAnchor
    e.Graphics.DrawLine(pen_set_ox, 0, CInt(PictureBox1.Height / 2), _
    CInt(PictureBox1.Width), CInt(PictureBox1.Height / 2))
'_____ Ось OY _____
    e.Graphics.DrawLine(pen_set_ox, CInt(PictureBox1.Width / 2), 0, _
    CInt(PictureBox1.Width / 2), CInt(PictureBox1.Width))
'_____ ПОДПИСИ ДАННЫХ _____
    e.Graphics.TranslateTransform(PictureBox1.Width / 2, _
    PictureBox1.Height / 2)
    For i = 0 To PictureBox1.Width Step 20
        e.Graphics.DrawString(CStr(i) / 10, _
        New Font("Times New Roman", 9, FontStyle.Italic), _
        Brushes.Green, New PointF(i, 0))
        e.Graphics.DrawString(-CStr(i) / 10, _
        New Font("Times New Roman", 9, FontStyle.Italic), _
        Brushes.Green, New PointF(0, i))
    Next
End Sub
#End Region
#Region "ПОБРОБУЙ СНАЧАЛА ПРОСТОЙ ПРИМЕР(ПОСТРОЕНИЕ СИНУСОИДЫ, _
или парабола, известного вам графика)"
Sub Button1_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button1.Click

```

```

Dim graphic As Graphics = PictureBox1.CreateGraphics
Dim demoPen As New Pen(Color.DarkBlue, 3)
'
Dim ppArray() As PointF
Dim k As Integer, y As Double, x As Double
For x = -600 To 500 Step 5
    ReDim Preserve ppArray(k) 'Если модификатор Preserve указан,
    'значения элементов копируются из существующего массива в новый.
    y = -50 * Math.Sin((1 / 20) * x)
    'y = -(1 / 100) * x ^ 2 REM парабола
    ppArray(k) = New PointF(x, y)
    k = k + 1
Next
graphic.DrawString("СИНУСОИДА", New Font("Times New Roman", 16, _
FontStyle.Bold), Brushes.Blue, New PointF(50, 300))
graphic.TranslateTransform(PictureBox1.Width / 2, _
PictureBox1.Height / 2)
' graphic.RotateTransform(90)
graphic.DrawCurve(demoPen, ppArray)
' graphic.DrawLines(Pens.Red, ppArray)
'graphic.DrawLine(Pens.DarkGreen, New Point(30, 40), New Point(50, 80))
'graphic.DrawPolygon(Pens.DarkGreen, ppArray)
'graphic.DrawClosedCurve(demoPen, ppArray)
End Sub
#End Region
#Region "Методы графического пользовательского интерфейса _
(ПОСТРОЕНИЕ 'МНОГОЛЕПЕСТКОВОЙ РОЗЫ')
Dim Разукрась
Sub Процедура_построения_кривой_вашего_варианта()
'Можете запускать по нажатию кнопки, т.е.:
'Private Sub Button2_Click(ByVal sender As System.Object, _
'ByVal e As System.EventArgs) Handles Button2.Click
    REM _____ P O Z A _____
    Dim graphic As Graphics = PictureBox1.CreateGraphics
    'Определяем PictureBox1, как поверхность рисования.
    Dim demoBrush As Brush 'Brush<Щетка, кисточка>
    Dim demoPen As New Pen(Color.Red, 2)
    'определяем кисть красного цвета для самой фигуры заданной толщины.
    Dim zalivka As New SolidBrush(Color.FromArgb(80, 255, 255, 50))
    'SolidBrush-<Solid-Rus.Сплошной>Определяет кисть одного цвета.
    'Кисти используются для заливки графических фигур, _
    'таких как прямоугольники, эллипсы, круги, многоугольники и контуры.
    Dim myTextureBrush As New TextureBrush(My.Resources.Chrysanthemum)
    'TextureBrush является объектом Brush, _
    'использующим изображение для заливки внутренней части
    Dim myHatchBrush As New HatchBrush(HatchStyle.DiagonalCross, _
Color.FromArgb(200, 0, 255, 100), Color.FromArgb(100, 255, 255, 1))
    'HatchBrush<Hatch-Rus.Люк>-Задаёт прямоугольную кисть со стилем
    'штриховки, основным цветом и цветом фона.
    demoBrush = myHatchBrush
    'переприсваивай demoBrush=>zalivka,myTextureBrush,myHatchBrush
    ЗАЛИВКА(Разукрась, zalivka, myTextureBrush, myHatchBrush)
    demoBrush = Разукрась
    'Кисти используются для заливки графических фигур, таких как
    'прямоугольники, эллипсы, круги, многоугольники и контуры
    '_____ Непосредственное построение кривой заданной в полярных координатах:

```

```

' RO=RO(FI), где FI ∈ [0,Pi] _____
Dim ppArray() As PointF
'PointF-представляет упорядоченную пару координат X и Y
'с плавающей запятой, определяющую точку на двумерной плоскости.
Dim k As Integer 'k-индекс одномерного массива ppArray()
'для занесения вычисляемых координат кривой листа задания
Dim x As Double, y As Double, RO As Double, FI As Double
graphic.TranslateTransform(PictureBox1.Width / 2, _
PictureBox1.Height / 2)
'Transform-<Рус.Преобразование>-Параллельный перенос
graphic.RotateTransform(alfa)
'- можно сделать поворот на [alfa] градусов
For FI = 0 To Math.PI Step 0.01
    ReDim Preserve ppArray(k + 1)
    k = k + 1
    'RO = (PictureBox1.Height / 2 - 30) * Sin(7 * FI)
    'Оставь эту строку, если радиус лепестка-ПОСТОЯННЫЙ
    RO = a * Sin(n * FI)
    x = RO * Cos(FI)
    y = -RO * Sin(FI)
    ppArray(k) = New PointF(x, y)
Next
'ЗАМЕЧАНИЕ:Фундаментальный сплайн представляет собой гладкую кривую,
'проходящую через заданные точки. Чтобы нарисовать фундаментальный сплайн,
'создайте объект Graphics и передайте адрес массива точек методу DrawCurve.
graphic.FillClosedCurve(demoBrush, ppArray)
'FillClosedCurve-<Fill-Ru.Полный><Curve-Ru.
'Кривая>заполняет(пером demoBrush) внутреннюю часть замкнутой
'фундаментальной кривой, определяемой массивом структур Point.
graphic.DrawLines(demoPen, ppArray)
'DrawLines-рисует набор сегментов линии (пером demoPen-граница),
'которые соединяют массив структур Point.
'_____ В РАБОТЕ МОЖНО ИСПОЛЬЗОВАТЬ СЛЕДУЮЩИЕ ВОЗМОЖНОСТИ:_____
'graphic.DrawCurve(Pens.Red, ppArray, 15)
'DrawCurve-строит фундаментальную кривую через точки
'указанного массива структур Point.
'ЗАМЕЧАНИЕ: Основной или фундаментальный сплайн – это
'последовательность отдельных кривых, объединенных в одну большую кривую.
'Сплайн задается массивом точек и параметром упругости
'Фундаментальный сплайн гладко (не образуя изломов и разрывов) проходит
'через каждую точку массива.
'graphic.DrawLine(Pens.DarkGreen, New Point(30, 40), New Point(50, 80))
'DrawLine-проводит линию, соединяющую две точки,
'задаваемые парами координат.
'graphic.DrawPolygon(Pens.DarkGreen, ppArray)
'DrawPolygon-рисует многоугольник, определяемый массивом структур Point.
graphic.DrawString("Поза", New Font("Times New Roman", 40), _
Brushes.Black, New PointF(50, 60))
'DrawString-создает указываемую текстовую строку в заданном месте
'с помощью определяемых объектов Brush и Font.
'graphic.DrawClosedCurve(demoPen, ppArray)
'DrawClosedCurve-строит замкнутую фундаментальную кривую,
'определяемую массивом структур Point.
End Sub
#End Region
#Region "Процедура переприсваивания заливки фона"

```

```

Sub ЗАЛИВКА(ByRef Разукрась, ByVal zalivka, ByVal myTextureBrush, _
    ByVal myHatchBrush)
'Параметры объявлены с ключевым словом ByRef=>
'в вызывающем коде можно обеспечить передачу аргумента
Select Case ComboBox1.Text
    Case "zalivka" 'Производим заливку прозрачным цветом
        Разукрась = zalivka
    Case "myTextureBrush"
        'Производим заливку 'картинкой' занесенной в Resources
        Разукрась = myTextureBrush
    Case "myHatchBrush"
        'Производим различного рода штриховку
        Разукрась = myHatchBrush
    Case Else
        Разукрась = zalivka
End Select
End Sub
#End Region
#Region "Динамическое изменение параметров кривых"
'_____ Изменение радиуса лепестка _____
Private Sub TrackBar1_Scroll(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles TrackBar1.Scroll
    TrackBar1.Maximum = 1000
    PictureBox1.Refresh() ' Refresh<Обновление>-
    'Принудительно создает условия, при которых элемент управления
    'делает недоступной свою клиентскую область и немедленно
    'перерисовывает себя и все дочерние элементы.
    a = TrackBar1.Value
    Процедура_построения_кривой_вашего_варианта()
    '_____ Количество лепестков _____
End Sub
Private Sub TrackBar2_Scroll(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles TrackBar2.Scroll
    TrackBar2.Maximum = 50
    PictureBox1.Refresh()
    n = TrackBar2.Value
    Процедура_построения_кривой_вашего_варианта()
End Sub
'_____ Поворот полученного изображения _____
Private Sub TrackBar3_Scroll(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles TrackBar3.Scroll
    TrackBar3.Maximum = 1000
    PictureBox1.Refresh()
    alfa = TrackBar3.Value
    Процедура_построения_кривой_вашего_варианта()
End Sub
#End Region
End Class

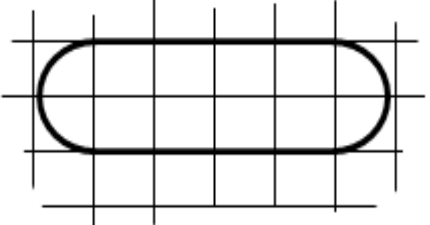
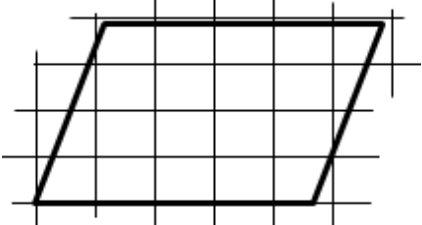
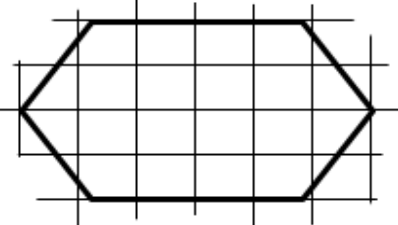
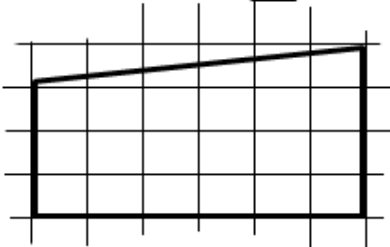
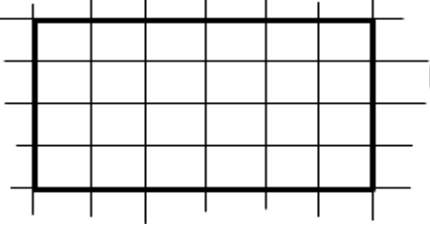

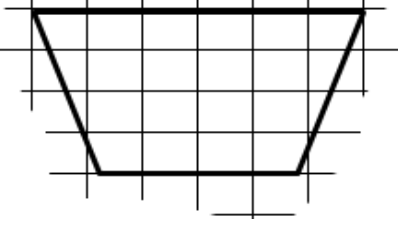
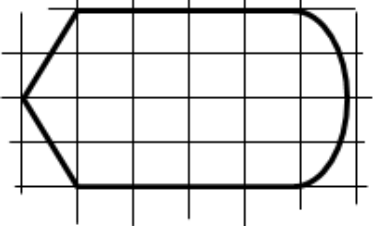
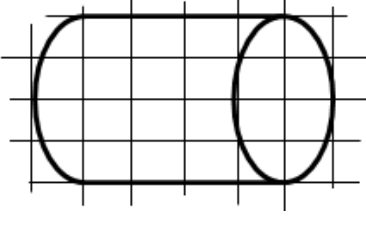
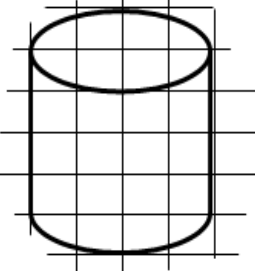
```

---

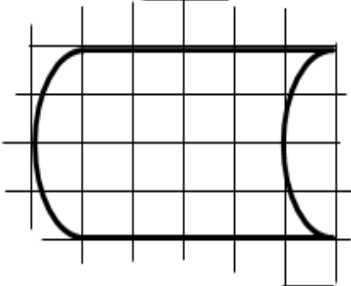
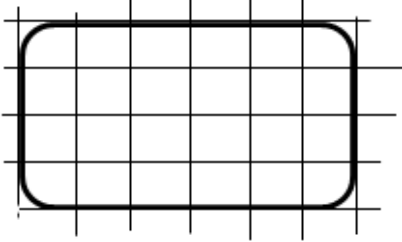
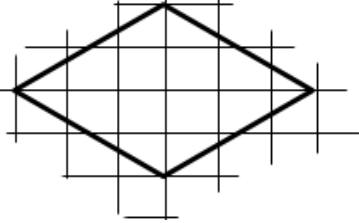
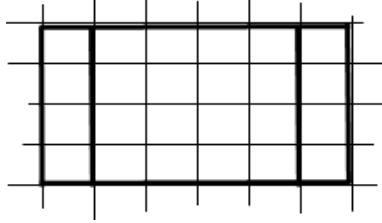
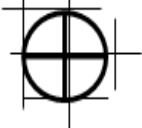
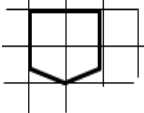
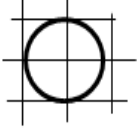
## ДОДАТОК А ДОДАТКОВІ ВІДОМОСТІ З VB

### А.1 Блок-схеми алгоритмів

Таблиця А.1 – Формати блок-схем

Початок / Кінець алгоритму	Дані
	
Циклічний процес	Ручне введення
	
Процес	Внутрішня пам'ять
	
Ручне керування	Дисплей
	
Пам'ять із прямим доступом	Магнітний диск
	

## Продовження таблиці А.1

Збережені дані		Альтернативний процес	
			
Умова		Типовий процес	
			
Вузол «Чи»	Посилання на іншу сторінку	Вузол	
			

## А.2 Терміни й поняття

Таблиця А.2 – Основні властивості VB

Властивість	Транскрипція	Переклад
1	2	3
BackColor 1) Back 2) Color	[bæk] ['klələ]	1) <b>задній</b> , зворотний 2) <b>колір</b> , красити, пофарбувати
BackColorImage 1) Background 2) Image	['bækgraund] ['imidʒ]	1) фон, задній план, <b>фон програми</b> ; 2) зображення, відображати, зображувати
BackColorImageLayout 1) Background 2) Image 3) Layout	['bækgraund] ['imidʒ] ['leiaut]	1) тло, задній план, <b>фон програми</b> ; 2) <b>зображення</b> , відображати, зображувати 3) <b>розміщення</b> , розташування, розмітка
Cursor	[kə:sə]	<b>курсор</b> , покажчик
FlatStyle 1) Flat 2) Style	[flæt] [stail]	1) плаский, <b>комірка</b> 2) <b>стиль</b> , тип, розробляти
Font	[fɒnt]	<b>шрифт</b>
ForeColor 1) Fore 2) Color	['klələ]	1) <b>передній</b> , одинарний 2) <b>колір</b> , красити, пофарбувати
Image	['imidʒ]	<b>зображення</b> , відображати, зображувати
ImageAlign 1) Image 2) Align	['imidʒ] [ə'lain]	1) <b>зображення</b> , відображати, зображувати 2) ставити, вирівнювати
ImageIndex 1) Image 2) Index	['imidʒ] ['indeks]	1) <b>зображення</b> , відображати, зображувати 2) <b>індекс</b> , стрілка, покажчик
ImageKey 1) Image 2) Key	['imidʒ] [ki:]	1) <b>зображення</b> , відображати, зображувати 2) <b>ключ</b> , клавіша
ImageList 1) Image 2) List	[list] ['imidʒ]	1) <b>зображення</b> , відображати, зображувати 2) <b>список</b> , перелік, аркуш, перерахування
RightToLeft 1) Right 2) To 3) Left	[raɪt] [tu] [left]	1) <b>правий</b> , 2) в, на, до 3) <b>лівий</b> , ліворуч
Text	[tekst]	<b>текст</b> , текстовий файл, оригінал
TextAlign 1) Text 2) Align	[tekst] [ə'lain]	1) <b>текст</b> , текстовий файл, оригінал 2) ставити, <b>вирівнювати</b>
TextImageRelation 1) Text 2) Image 3) Relation	[tekst] ['imidʒ] [rɪ'leɪʃn]	1) <b>текст</b> , текстовий файл, оригінал 2) <b>зображення</b> , відображати, зображувати 3) відношення, співвідношення, <b>зв'язок</b>
AutoSize 1) Auto 2) Size	[saɪz]	1) <b>автоматичний</b> 2) <b>розмір</b> , розміром
AutoSizeMode 1) Auto 2) Size 3) Mode	[saɪz] [məʊd]	1) <b>автоматичний</b> 2) <b>розмір</b> , розміром 3) спосіб, метод, <b>режим</b>
Location	[ləu'keɪʃən]	<b>розташування</b> , положення
Size	[saɪz]	<b>розмір</b> , розміром
ContextMenu 1) Context 2) Menu	['kɒntekst] ['menju:]	1) <b>контекст</b> ; 2) <b>меню</b>
DialogResult 1) Dialog 2) Result	['daɪələʒ] [rɪ'zʌlt]	1) <b>діалог</b> , розмова 2) закінчуватися, приводити, <b>результат</b>
Tag	[tæg]	ярлик

Таблиця А.3 – Основні події VB

Подія	Транскрипція	Переклад
1	2	3
Anchor	[ˈæŋkə]	<b>установитися</b> , закріплювати
Dock	[dɒk]	<b>скорочувати</b> , урізувати
Activate	[ˈæktɪveɪt]	<b>активувати</b> , активізувати
AutoSizeChange 1) Auto 2) Size 3) Change	[saɪz] [tʃeɪndʒ]	1) <b>автоматичний</b> 2) <b>розмір</b> , розміром 3) <b>змінювати</b> , замінити, обмінювати
AutoValidateChange 1) Auto 2) Validate 3) Change	[ˈvælɪdeɪt] [tʃeɪndʒ]	1) <b>автоматичний</b> 2) затверджувати, <b>ратифікувати</b> 3) <b>змінювати</b> , замінити, обмінювати
BackColorChanged 1) Back 2) Color 3) Change	[bæk] [ˈklɒlə] [tʃeɪndʒ]	1) <b>задній</b> , зворотний 2) <b>колір</b> , красити, пофарбувати 3) змінювати, замінити, обмінювати
BackgroundImageChanged 1) Background 2) Image 3) Change	[ˈbækgraʊnd] [ˈɪmɪdʒ] [tʃeɪndʒ]	1) фон, задній план, тло програми 2) зображення, відображати 3) змінювати, замінити, обмінювати
BackgroundImageLayoutChanged 1) Background 2) Image 3) Layout 4) Change	[ˈbækgraʊnd] [ˈɪmɪdʒ] [ˈleɪaʊt] [tʃeɪndʒ]	1) фон, задній план, тло програми ; 2) зображення, відображати, зображувати 3) розміщення, розташування, розмітка 4) змінювати, замінити, обмінювати
BindingContextChanged 1) Binding 2) Context 3) Change	[ˈbaɪndɪŋ] [ˈkɒntekst] [tʃeɪndʒ]	1) зв'язок, плетіння, скріплення; 2) контекст; 3) змінювати, замінити, обмінювати
CausesValidationChanged 1) Causes 2) Validate 3) Change	[kəʊz] [ˈvælɪdeɪt] [tʃeɪndʒ]	1) причини 2) затверджувати, ратифікувати 3) змінювати, замінити, обмінювати
ChangeUICues 1) Change 2) Cues	[tʃeɪndʒ] [kjuː]	1) змінювати, замінити, обмінювати 2) репліка, сигнал
ClientSizeChanged 1) Client 2) Size 3) Change	[ˈklaɪənt] [saɪz] [tʃeɪndʒ]	1) клієнт, замовник, користувач 2) розмір, розміром 3) змінювати, замінити, обмінювати
Contextmenuchanged 1) Context 2) Menu 3) Change	[ˈkɒntekst] [ˈmenjuː] [tʃeɪndʒ]	1) контекст; 2) меню; 3) змінювати, замінити, обмінювати
ContextMenuStripChanged 1) Context 2) Menu 3) Strip 4) Change	[ˈkɒntekst] [ˈmenjuː] [stri:p] [tʃeɪndʒ]	1) контекст; 2) меню; 3) панель, вузька смуга, невелика ділянка 4) змінювати, замінити, обмінювати
ControlAdded 1) Control 2) Add	kenˈtrɒl] [æd]	1) контроль, контрол, керування, боротьба 2) додавати, складати
ControlRemoved 1) Control 2) Remove	[kenˈtrɒl] [rɪˈmuːv]	1) контроль, контрол, керування, боротьба 2) видаляти, знімати, забирати
CursorChanged 1) Cursor 2) Change	[kəːsə] [tʃeɪndʒ]	1) курсор, покажчик 2) змінювати, замінити, обмінювати
Deactivate		дезактивувати
Disposed	[dɪˈspəʊz]	розташований
DockChanged 2) Change	[tʃeɪndʒ]	2) змінювати, замінити, обмінювати
DoubleClick 1) Double 2) Click	[ˈdʌbl] [kɪk]	1) подвоювати, подвійний 2) натисніть, клацати, клацання



## Продовження таблиці А.3

1	2	3
DragDrop 1) Drag 2) Drop	[dræg] [drɒp]	1) тягтися, тягти 2) падіння, зниження
DragEnter 1) Drag 2) Enter	[dræg] ['entə]	1) тягтися, тягти 2) уводити, входити, проникати, вносити
DragLeave 1) Drag 2) Leave	[dræg] [li:v]	1) тягтися, тягти 2) залишати, іти, їхати, припиняти
DragOver 1) Drag 2) Over	[dræg] ['əʊvə(r)]	1) тягтися, тягти 2) над, за, по
EnabledChanged 1) Enabled 2) Change	[ɪ'neɪb(ə) l] [tʃeɪndʒ]	1) доступно 2) змінювати, замінювати, обмінювати
Enter	['entə]	уводити, входити, проникати, вносити
FontChanged 1) Font 2) Change	[fɒnt] [tʃeɪndʒ]	1) шрифт 2) змінювати, замінювати, обмінювати
ForeColorChanged 1) Fore 2) Color 3) Change	['kɒlə] [tʃeɪndʒ]	1) передній, одинарний 2) колір, красити, пофарбувати 3) змінювати, замінювати, обмінювати
FormCosed 1) Form 2) Closed	[fɔ:m] [kləʊz]	1) форма, бланк, клас 2) закривати, містити
Formclosing 1) Form 2) Closing	[fɔ:m] [kləʊz]	1) форма, бланк, клас 2) закривати, містити
HelpButtonClicked 1) Help 2) Button 3) Clicked	[help] [bʌtn] [klik]	1) допомога, сприяння, підказка 2) кнопка, гудзик 3) натисніть, клацати, клацання
HelpRequested 1) Help 2) Request	[help] [rɪ'kwest]	1) допомога, сприяння, підказка 2) запит, прохання
ImeModeChanged 2) Mode 3) Changed	[məʊd] [tʃeɪndʒ]	2) спосіб, метод, нахилення 3) змінювати, замінювати, обмінювати
InputLanguageChanged 1) Input 2) Language 3) Changed	['ɪnpʊt] ['læŋgwɪdʒ] [tʃeɪndʒ]	1) уводити, уведення, вхідний 2) мова, формулювання 3) змінювати, замінювати, обмінювати
InValidated	[ɪn'vælɪdeɪt]	недійсні
KeyDown 1) Key 2) Down	[ki:] [daʊn]	1) ключ, клавіша 2) унизу, униз, опускати
KeyUp 1) Key 2) Up	[ki:] [ʌp]	1) ключ, клавіша 2) нагору, вище, нагорі, наверх;
Layout	[leɪ]	макет, розташування, розкладка
Leave	[li:v]	залишати, іти, їхати, припиняти
LocationChanged 1) Location 2) Changed	[ləu'keɪʃən] [tʃeɪndʒ]	1) розташування, місцезнаходження, положення 2) змінювати, замінювати, обмінювати
LostFocus 1) Lost 2) Focus	[lu:z] ['fəʊkəs]	1) загублений 2) сфокусувати, фокусувати
MaximumSizeChanged 1) Maximum 2) Size 3) Changed	['mæksɪmə] [saɪz] [tʃeɪndʒ]	1) максимальний, вищий 2) розмір, розміром 3) змінювати, замінювати, обмінювати
MenuComplete 1) Menu 2) Complete	['menju:] [kəm'plɪt]	1) меню; 2) повний, зроблений
MenuStart 1) Menu 2) Start	['menju:] [stɑ:t]	1) меню; 2) початок, запуск, старт

## Закінчення таблиці А.3

1	2	3
MinimumSizeChanged 1) Minimum 2) Size 3) Changed	[stɑ:t] [saiz] [tʃeɪndʒ]	1) мінімальний 2) розмір, розміром 3) змінювати, замінити, обмінювати
MouseClicked 1) Mouse 2) Click	[maʊs] [klik]	1) миша, мишка 2) натисніть, клацати, клацання
MouseDoubleClick 1) Mouse 2) Double 3) Click	[maʊs] [ˈdʌbl] [klik]	1) миша, мишка 2) подвоювати, подвійний 3) натисніть, клацати, клацання
MouseDown 1) Mouse 2) Down	[maʊs] [daʊn]	1) миша, мишка 2) унизу, униз, опускати
MouseEnter 1) Mouse 2) Enter	[maʊs] [ˈentə]	1) миша, мишка 2) уводити, входити, проникати, вносити
MouseHover 1) Mouse 2) hover	[maʊs] [ˈhɒvə(r) ]	1) миша, мишка 2) зависати, коливатися, вертітися
MouseLeave 1) Mouse 2) Leave	[maʊs] [li:v]	1) миша, мишка 2) залишати, іти, їхати, припиняти
MouseMove 1) Mouse 2) Move	[maʊs] [mu:v]	1) миша, мишка 2) рухати, рухатися
MouseUp 1) Mouse 2) Up	[maʊs] [ʌp]	1) миша, мишка 2) нагору, вище, нагорі, наверх;
MouseWheel 1) Mouse 2) Wheel	[maʊs] [wi:l]	1) миша, мишка 2) колесо, коліща
Move	[mu:v][	2) рухати, рухатися
PaddingChanged 1) Padding 2) Changed	[ˈpædɪŋ] [tʃeɪndʒ]	1) набивання, ґрунтування 2) змінювати, замінити, обмінювати
ParentChanged 1) Parent 2) Changed	[ˈpeərənt] [tʃeɪndʒ]	1) батьківський, вихідний 2) змінювати, замінити, обмінювати
PreviewKeyDown 1) Preview 2) Key 3) Down	[ˈpri:ˈvju:] [ki:] [daʊn]	1) попередній перегляд, анонс 2) ключ, клавіша 3) унизу, униз, опускати
Regionchanged 1) Region 2) Changed	[ˈri:dʒ(ə) n] [tʃeɪndʒ]	1) регіон, область, район 2) змінювати, замінити, обмінювати
Resize		змінити розмір
ResizeBegin 1) Resize 2) Begin	[bɪˈɡɪn]	1) змінити розмір 2) починати
ResizeEnd 1) Resize 2) End	[end]	1) змінити розмір 2) кінець, закінчення, завершення
Show	[ʃəʊ]	показувати, демонструвати
SizeChanged 1) Size 2) Changed	saiz] [tʃeɪndʒ]	1) розмір, розміром 2) змінювати, замінити, обмінювати
StyleChanged 1) Style 2) Changed	[stɑɪl] [tʃeɪndʒ]	1) стиль, тип, розробляти 2) змінювати, замінити, обмінювати
SystemColorsChanged 1) System 2) Color 3) Changed	[ˈsɪstɪm] [ˈkɒlə] [tʃeɪndʒ]	1) система, організм 2) колір, красити, пофарбувати 3) змінювати, замінити, обмінювати
TextChanged 1) Text 2) Changed	[tekst] [tʃeɪndʒ]	1) текст, текстовий файл, оригінал; 2) змінювати, замінити, обмінювати
Validate	[ˈvælɪdeɪt]	затверджувати
VisibleChanged 1) Visible 2) Changed	[ˈvɪzɪb(ə) l] [tʃeɪndʒ]	1) видимий 2) змінювати, замінити, обмінювати

Таблиця А.4 – Основні програмні конструкції VB

Конструкція	Транскрипція	Переклад	Приклад
1	2	3	4
<b>End</b>	[end]	<b>кінець</b> , закінчення, завершення	End Class – завершення класу
<b>Dash</b>	[dæʃ]	1) риска, ривок, тире	Dashstyle.Dash – стиль пунктирних ліній. Пунктир
<b>Add</b>	[æd]	додавати, складати	Columns.Add – стовпці додати
<b>Addextension</b> 1) Add 2) Extension	[æd] [ɪks'tenʃən]	1) додавати, складати 2) розширення, продовження, додатковий номер, розширення імені файлу	Addextension = True – Додати розширення= Істина
<b>Allowusertoaddrows</b> 1) Allow 2) User 3) To 4) Add 5) Row	[ə'laʊ] ['ju:zə] [æd]	1) дозволяти, надавати 2) користувач, користувачький, споживач 3) до, для, в, на, до 4) додавати, складати 5) рядок, ряд	Allowusertoaddrows – Указуємо, чи відображається для користувача параметр додавання рядків
<b>As</b>	[æz]	як, згідно, наприклад	sender As System.Object – відправник як системний об'єкт
<b>Autosizecolumnsmode</b> 1) Auto 2) Size 3) Column 4) Mode	[saɪz] ['kɒləm] [məʊd]	1) автоматичний 2) розмір, розміром 3) колонка, стовпчик, <b>стовпець</b> 4) спосіб, метод	Autosizecolumnsmode – повертає або задає значення, що вказує, як визначається ширина стовпця
<b>BackColor</b> 1) Back 2) Color	[bæk] ['kɒlə]	1) задній, зворотний 2) колір, красити, пофарбувати	BackColor – задній колір
<b>Backgroundworker</b> 1) Background 2) Worker	[ˈbækgraʊnd] [ˈwɜ:kə]	1) фон, задній план, <b>фон програми</b> ; 2) працівник, <b>робітник</b> , співробітник	виконує операцію в окремому потоці
<b>Bindingnavigator</b> 1) Binding 2) Navigator	[ˈbaɪndɪŋ] [ˈnævɪɡeɪtə]	1) зв'язок, плетіння, скріплення; 2) <b>навігатор</b>	указує інтерфейс користувача для навігації
<b>Bindingsource</b> 1) Binding 2) Source	[ˈbaɪndɪŋ] [sɔ:s]	1) зв'язок, плетіння, скріплення; 2) <b>джерело</b> , вихідний код	забезпечує здатність навігації, фільтрації, сортування й відновлення
<b>Brush</b>	[brʌʃ]	малювати кистю, <b>кисть</b>	Brushes.Green – Кисть. Зелений
<b>Button</b>	[bʌtn]	<b>кнопка</b> , гудзик	click button – натисніть кнопку
<b>Call</b>	[kɔ:l]	називати, <b>викликати</b> , кликати, повідомляти	Call підпрограма (m, t, x, y, z) – викликаємо підпрограму й передаємо їй у якості аргументів змінні: m, t, x, y, z
<b>Chart</b>	[tʃɑ:t]	<b>діаграма</b> , графік, таблиця, схема, креслення	елемент керування діаграми
<b>Checkbox</b>	[tʃek][bɒks]	прапорець	Checkbox дозволяє користувачеві додати або вилучити відповідний параметр

## Продовження таблиці А.4

1	2	3	4
<b>Checkedlistbox</b> 1) Checked 2) List 3) Box	[tʃeked] [list] [bɒks]	<b>1) перевірене</b> , зупинений, стриманий 2) <b>список</b> , перелік, аркуш, перерахування; 3) коробка, ящик, бокс, <b>вкладка</b>	відображає список елементів із кнопками у вигляді прапорців ліворуч кожного елемента
<b>Class</b>	[kla:s]	<b>клас</b> , розряд, категорія, сорт	Public Class – відкритий клас
<b>Clear</b>	[klɪə]	<b>очищати</b> , покривати, чистити, <b>чіткий</b>	
<b>Click</b>	[klik]	<b>натисніть</b> , клацати, клацання, клацання	MenuItem_Click – натисніть пункт меню
<b>Color</b>	[ˈkɒlə]	<b>колір</b> , красити, пофарбувати	BackColor – задній колір
<b>Colordialog</b> 1) Color 2) Dialog	[ˈkɒlə] [ˈdaɪələg]	1) <b>колір</b> , колорит, відтінок 2) <b>діалог</b> , розмова	відображає доступні кольори й елементи правління
<b>Column</b>	[ˈkɒləm]	стовпчик, стовпчик, <b>стовпець</b>	Columns.Add – стовпці додати
<b>Columncount</b> 1) Column 2) Count	[ˈkɒləm] [ˌkaʊnt]	1) стовпчик, стовпчик, <b>стовпець</b> 2) уважати, поррахувати, рахунок, <b>кількість</b>	повертає або задає число стовпців, що відображаються в об'єкті DataGridView.
<b>Combobox</b>	[kɒmbɒ] [bɒks]	Поле зі списком	відображає редаговане текстове поле й розкриває список значень, що редагуються
<b>Computer</b>	[kəmˈpjʊ:tə]	<b>комп'ютер</b> , ЕОМ	My.Computer – Мій.Комп'ютер
<b>Contextmenustrip</b> 1) Context 2) Menu 3) Strip	[ˈkɒntekst] [ˈmenju:] [stri:p]	<b>1) контекст</b> ; 2) <b>меню</b> ; 3) <b>панель</b> , вузька смуга, невелика ділянка	відображає меню швидкого доступу
<b>Creategraphics</b> 1) Create 2) Graphics	[kri:'eɪt] [ˈgræfɪk]	1) <b>створювати</b> , викликати, задумувати 2) <b>графіка</b> , графічний символ	Creategraphics – створити графіку
<b>Dashstyle</b> 1) Dash 2) Style	[dæʃ] [staɪl]	1) <b>риска</b> , ривок, тире 2) <b>стиль</b> , тип, розробляти	задає стиль пунктирних ліній
<b>Datagridveiw</b> 1) Data 2) Grid 3) View	[ˈdeɪtə] [ˈgrɪd] [vju:]	1) <b>дані</b> , інформація, показники, відомості; 2) <b>сітка</b> , розв'язок; 3) огляд, <b>вид</b> , думка, погляд, проекція;	відображає рядки й стовпці даних у сітці
<b>Datagridviewcellstyle</b> 1) Data 2) Grid 3) View 4) Cell 5) Style	[ˈdeɪtə] [ˈgrɪd] [vju:] [sel] [staɪl]	1) <b>дані</b> , інформація, показники, відомості; 2) <b>сітка</b> , розв'язок; 3) огляд, <b>вид</b> , думка, погляд, проекція; 4) <b>комірка</b> , клітка, <b>елемент</b> 5) <b>стиль</b> , тип	Datagridviewcellstyle – подає відомості про форматування й стиль, застосовувані до окремих комірок в елементі керування DataGridView
<b>Dataset</b> 1) Data 2) Set	[ˈdeɪtə] [set]	1) <b>дані</b> , інформація, показники, відомості; 2) <b>установлювати</b> , задавати, ставити, набір, комплект, комплекс	представляє кеш даних у пам'яті

## Продовження таблиці А.4

1	2	3	4
<b>Datetimepicker</b> 1) Pick 2) Time 3) Date	[pik] [taim] [deit]	1) вибір дати й часу; 2) <b>вибір</b> , удар, зубочистка; 3) <b>час</b> , часи, раз, момент, строк; 4) <b>дата</b> , день, строк, час	дозволяє вибрати час і дату й відобразити їх в обраному форматі
<b>Defaultcellstyle</b> 1) Default 2) Cell 3) Style	[dr'fɔ:lt] [sel] [stail]	1) <b>за замовчуванням</b> , ухвалювати значення за замовчуванням, значення за замовчуванням 2) комірка, клітка, <b>елемент</b> 3) <b>стиль</b> , тип	Datagridview.Defaultcellstyle – Властивість, що повертає або задає стиль комірки за замовчуванням, яке буде застосовуватися до комірок в об'єкті Datagridview1, якщо не задані які-небудь інші властивості стилю комірки
<b>Defaulttext</b> 1) Default 2) Ext	[dr'fɔ:lt]	1) <b>за замовчуванням</b> , ухвалювати значення за замовчуванням, значення за замовчуванням 2) <b>розширення</b>	Defaulttext = розширення за замовчуванням=
<b>Dim</b>	[dim]	потьмяніти, недоступний, <b>вимір</b>	Dim x As Double, y As Double, z As Double – стандартний опис [Dim] усіх використовуваних елементів у наступному коді
<b>Directoryentry</b> 1) Directory 2) Entry	[di'rektəri] ['entri]	1) <b>каталог</b> , директорія, папка, дирекція 2) вступ, запис, вхід, в'їзд, <b>уведення</b>	инапсулює вузол або об'єкт ієрархії Active Directory
<b>Double</b>	['dʌbl]	подвоювати, <b>подвійний</b>	tn As Double – tn Як число подвійної точності, із плаваючої коми
<b>Drawcurve</b> 1) Draw 2) Curve	[drɔ:] [kə:v]	1) <b>малювати</b> , залучати, одержувати 2) <b>крива</b> , траєкторія	Drawcurve – малювати криву
<b>Drawing</b>	['drɔ:ɪŋ]	малюнок, <b>малювання</b>	Drawing.Font – Малювання.Шрифт
<b>Drawline</b> 1) Draw 2) Line	[drɔ:] [laɪn]	1) <b>малювати</b> , залучати, одержувати 2) <b>лінія</b> , черга, рядок	Drawline – малювати лінію
<b>Drawstring</b> 1) Draw 2) String	[drɔ:] [strɪŋ]	1) <b>малювати</b> , залучати, одержувати 2) послідовність, ряд, ланцюг, <b>рядок</b>	Drawstring – Малювати рядок
<b>Elementhost</b> 1) Element 2) Host	['elɪmənt] [hɒst]	1) <b>елемент</b> , стихія, фактор, ланка 2) <b>множина</b> , ведучий	множина елементів
<b>Errorprovider</b> 1) Error 2) Provider	['erə] [prə'vaɪdə]	1) <b>помилка</b> , погрішність, омана, відхилення 2) постачальник, <b>провайдер</b>	надає інтерфейс користувача, що вказує користувачеві про наявність помилки в системі керування формою

## Продовження таблиці А.4

1	2	3	4
<b>Eventargs</b> 1) Event 2) Args	1) [ˈi:vən] 2) [ˈɑ:g]	1) випадок, подія, результат, <b>результат</b> ; 2) аргументи	Eventargs – результат аргументу
<b>Eventlog</b> 1) Event 2) Log	[iˈvent] [lɒg]	1) подія, випадок, подія, результат, <b>результат</b> 2) <b>логарифм</b> , записувати, логарифмічний	забезпечує взаємодія з журналами подій Windows
<b>Exclamation</b>	[ˌɛkskləˈmeɪʃən]	вигук	
<b>False</b>	[fɔ:ls]	<b>неправда</b> , неправильний	Visible = False – Видимий=Неправда
<b>Filename</b> 1) File 2) Name	[faɪl] [neɪm]	1) <b>файл</b> , досьє, архів, справа, зберігати, реєструвати 2) <b>ім'я</b> , назва, називати	Filename – ім'я файлу
<b>Filesystem</b> 1) File 2) System	[faɪl] [ˈsɪstɪm]	1) <b>файл</b> , досьє, архів, справа, зберігати, реєструвати 2) <b>система</b> , організм	My.Computer.Filesystem – Мій.Комп'ютер. Файлова система
<b>Filesystemwatcher</b> 1) File 2) System 3) Watcher	[[faɪl] [ˈsɪstɪm] [ˈwɒtʃə]	1) <b>файл</b> , досьє, архів, справа, зберігати, реєструвати 2) <b>система</b> , організм 3) <b>спостерігач</b> , дослідник	стежить за повідомленнями про зміну у файловій системі й створює події, коли змінюється каталог або файл
<b>Filter</b>	[ˈfɪltə]	фільтрувати, <b>фільтр</b>	Filter = – фільтр=
<b>Folderbrowserdialog</b> 1) Folder 2) Browser 3) Dialog	[ˈfouldə] [brauzə] [ˈdaɪələg]	1) <b>папка</b> , швидкозшивач, фальцювальник 2) <b>браузер</b> 3) <b>контроль</b> , керування, боротьба	відображає діалогове вікно, що дозволяє користувачеві вибрати папку
<b>Font</b>	[fɒnt]	шрифт	Drawing.Font – Малювання.Шрифт
<b>Fontdialog</b> 1) Font 2) Dialog	[fɒnt] [ˈdaɪələg]	1) <b>шрифт</b> , джерело 2) <b>діалог</b> , розмова	відображає діалогове вікно, що дозволяє користувачу обрати шрифт зі списку шрифтів
<b>Fontstyle</b> 1) Font 2) Style	[fɒnt] [stɑɪl]	1) <b>шрифт</b> 2) <b>стиль</b> , тип, розробляти	Fontstyle – стиль шрифту
<b>For</b>	[fɔ:]	для, для того, щоб, за	For t = tn To tk Step th – для t = tn До tk крок th
<b>Forecolor</b> 1) Fore 2) Color	[ˈkɒlə]	1) <b>передній</b> , одинарний 2) колір, красити, пофарбувати	Forecolor – передній колір
<b>Form</b>	[fɔ:m]	<b>форма</b> , бланк, клас	Windows.Forms. – форми Windows
<b>Format</b>	[ˈfɔ:mæt]	<b>форматувати</b> , отформатувати, формат	format As – форматувати як
<b>Gridcolor</b> 1) Grid 2) Color	[ˈgrɪd] [ˈkɒlə]	1) <b>сітка</b> , розв'язок; 2) <b>колір</b> , красити, пофарбувати	Gridcolor = Color.Red Повертає або задає колір ліній сітки, що розділяють гнізда.
<b>GroupBox</b> 1) Group 2) Box	[gru:p] [bɒks]	1) <b>група</b> , групувати, згрупувати 2) бокс, <b>вкладка</b> , коробка, ящик	GroupBox1.Visible = False – ГрупаВкладок1. Видима =Неправда

## Продовження таблиці А.4

1	2	3	4
<b>Handle</b>	[ˈhændəl]	<b>управляти</b> , регулювати, ручка, рукоять	Handles Textbox1 – керування текстовою вкладкою
<b>Hatchbrush</b> 1) Hatch 2) Brush	[hætʃ] [brʌʃ]	1) <b>штрихування</b> , грати 2) малювати кистю, <b>кисть</b>	Hatchbrush – Задає прямокутну кисть зі стилем штрихування, основним кольором і кольором тла.
<b>Headertext</b> 1) Header 2) Text	[ˈhedə] [tekst]	1) <b>рядок</b> , ряд 2) <b>текст</b> , текстовий файл, оригінал;	Headertext – рядок тексту
<b>Height</b>	[haɪt]	<b>висота</b> , ріст	Height Step 20 – Висота Крок 20
<b>Helpprovider</b> 1) Help 2) Provider	[help] [prəˈvaɪdə]	1) <b>допомога</b> , сприяння, підказка 2) постачальник, <b>провайдер</b>	забезпечує для елементів керування спливаюче вікно довідки або вікно оперативної довідки
<b>If</b>	[ɪf]	<b>якщо</b> , чи	
<b>Imagelist</b> 1) Image 2) List	[list] [ˈɪmɪdʒ]	1) <b>зображення</b> , відобразити, зображувати 2) <b>список</b> , перелік, аркуш, перерахування	управління колекцією зображень, які звичайно використовуються іншими елементами керування
<b>Import</b>	[ɪmˈpɔ:t]	<b>імпортувати</b> , вносити	Imports System.Math – Імпортувати Система.Математика
<b>Information</b>	[ˌɪnfəˈmeɪʃən]	інформація, інформаційний	
<b>Inputbox</b> 1) Input 2) Box	[ˈɪnpʊt] [bɒks]	1) <b>уводити</b> , введення, вхідний 2) бокс, <b>вкладка</b> , коробка, ящик	Inputbox – уведіть вкладку
<b>Integer</b>	[ˈɪntɪdʒə]	<b>ціле число</b> , цілочисельний	As Integer – як ціле число
<b>Item</b>	[ˈaɪtəm]	<b>пункт</b> , параграф, стаття	MenuItem – пункт меню
<b>Label</b>		етикетка, мітка, <b>ярлик</b> , наклейка, маркування, бирка	надає елементу керування ткст опису або на інформацію під час виконання
<b>Linklabel</b> 1) Link	[lɪŋk]	<b>посилання</b> , зв'язок, лінія зв'язку, з'єднувати	відображає елемент керування «напис», підтверджує форматування.
<b>Listbox</b> 1) List 2) Box	[list] [bɒks]	1) <b>список</b> , перелік, аркуш, перерахування; 2) коробка, ящик, бокс, <b>вкладка</b>	відображає список, з якого користувач може вибрати елементи
<b>Listview</b> 1) View 2) List	[vju:] [list]	1) огляд, <b>вид</b> , думка, погляд, проекція; 2) <b>список</b> , перелік, аркуш, перерахування	відображає колекцію елементів в одному з п'яти різних вистав
<b>Load</b>	[ləʊd]	завантажувати, заряджати, <b>завантаження</b> , вантажитися	Handles Me.Load – керування Мені.Завантаження
<b>Location</b>	[ləʊˈkeɪʃən]	<b>розташування</b> , місцезнаходження, положення	Me.Location – Мені. Розташування

## Продовження таблиці А.4

1	2	3	4
<b>MaskedTextBox</b> 1) Masked 2) Text 3) Box	[ma:skt] [tekst] [bɒks]	1) замаскований, маскований, <b>схований</b> ; 2) <b>текст</b> , текстовий файл, оригінал; 3) коробка, ящик, бокс, <b>вкладка</b>	використовує маску для розрізнення припустимого й неприпустимого введення даних користувачем
<b>Math</b>		<b>математика</b> , математичний	Imports System.Math – імпортувати Система.Математика
<b>max</b>		максимальний	
<b>Me</b>		мене, <b>мені</b>	Handles Me.Load – керування Мені.Завантаження
<b>Menu</b>	[ˈmenju:]	<b>меню</b>	MenuItem – пункт меню
<b>MenuStrip</b> 1) Menu 2) Strip	[ˈmenju:] [stri:p]	1) <b>меню</b> ; 2) <b>панель</b> , вузька смуга, невелика ділянка	відображає команди додатка й параметри, що згруповані за функціональністю
<b>MessageQueue</b> 1) Message 2) Queue	[ˈmesidʒ] [ˈkju:]	1) <b>повідомлення</b> , лист 2) <b>черга</b> , хвіст	надає доступ до черги сервера черги повідомлень
<b>min</b>		мінімальний	
<b>MonthCalendar</b> 1) Month 2) Calendar	[mʌnθ] [ˈkælɪndə]	<b>1) місяць</b> ; 2) <b>календар</b> , список, реєструвати	відображає місячний календар, за допомогою якого користувач може вибрати дату
<b>MouseEnter</b> 1) Mouse 2) Enter	[maʊs] [ˈentə]	1) миша, <b>мишка</b> 2) <b>уводити</b> , входити, проникати, вносити	TextBox1_Mouseenter – Текстова вкладка_уведення мишкою
<b>MouseLeave</b> 1) Mouse 2) Leave	[maʊs] [li:v]	1) миша, <b>мишка</b> 2) залишати, іти, їхати, <b>припиняти</b>	TextBox1.Mouseleave – Текстова вкладка_припинити мишку
<b>MsgBoxStyle</b> 1) Msg 2) Box 3) Style	[bɒks] [stɑɪl]	1) <b>повідомлення</b> 2) бокс, <b>вкладка</b> , коробка, ящик 3) стиль, тип, <b>розробляти</b>	
<b>MyBase</b> 1) My 2) Base	[maɪ] [beɪs]	1) мій, <b>моя</b> , моє 2) основа, <b>база</b>	MyBase.Load – МояБаза.Завантаження
<b>New</b>	[nju:]	<b>новий</b> , оновлений	New Point – нова крапка
<b>NotifyIcon:</b> 1) Notify 2) Icon	[ˈnɒtɪfaɪ] [ˈaɪkɒn]	1) сповіщати, <b>повідомляти</b> , інформувати, повідомляти; 2) зображення, значок, <b>іконка</b>	відображає позначку в області повідомлень праворуч від панелі задач
<b>NumericUpDown</b> 1) Numeric 2) Up 3) Down	[nju:ˈmerɪkəl] [ʌp] [daʊn]	<b>1) числовий</b> , цифровий; 2) <b>нагору</b> , вище, нагорі, наверх; 3) унизу, <b>униз</b> , опускати	відображає єдине числове значення, яке користувач може збільшити або зменшити за допомогою «вгору – униз» керування елементами
<b>Object</b>		<b>об'єкт</b> , предмет, мета, намір	System.Object – системний об'єкт



## Продовження таблиці А.4

1	2	3	4
<b>Opacity</b>	[əu'pæsɪtɪ]	<b>непрозорість</b> , помутніння	Opacity = 100 – непрозорість=100
<b>Openfiledialog</b> 1) Open 2) File 3) Dialog	[ˈoʊpən] [faɪl] [ˈdaɪəlɒɡ]	1) <b>відкрити</b> , відкривати, відкриватися 2) <b>файл</b> , досьє, архів, справа, зберігати, реєструвати 3) <b>діалог</b> , розмова	відображає діалогове вікно, що дозволяє користувачеві відкрити файл
<b>Pagesetupdialog</b> 1) Page 2) Setup 3) Dialog	[peɪdʒ] [ˈdaɪəlɒɡ]	1) <b>сторінка</b> , смуга 2) <b>установка</b> , обладнання, система 3) <b>діалог</b> , розмова	відображає діалогове вікно, що дозволяє користувачеві змінювати параметри сторінки для печатки, включаючи поля й орієнтацію паперу
<b>Paint</b>	[peɪnt]	<b>малювати</b> , красити, описувати	Handles Me.Paint – керування Мені.Малювати
<b>Painteventargs</b> 1) Paint 2) Event 3) Args	[peɪnt] [ˈi:vən] [ˈɑ:g]	1) <b>малювати</b> , красити, описувати 2) випадок, подія, результат, <b>результат</b> ; 3) <b>аргументи</b>	
<b>Pen</b>	[pen]	ручка	As New Pen – як нова ручка
<b>pen_set</b> 1) pen 2) set	[pen] [set]	1) <b>ручка</b> 2) <b>група</b> , набір, <b>установлювати</b>	
<b>Perfomancecounter</b> 1) Perfomance 2) Counter	[pə'fɔ:məns] [ˈkaunte]	1) <b>виконання</b> , виконання, дія 2) <b>лічильник</b> , вимірник, обчислювальне обладнання	представляє компонент лічильника продуктивності Windows
<b>Picturebox</b> 1) Picture 2) Box	[ˈpɪktʃə] [bɒks]	1) картинка, картина, <b>зображення</b> , малюнок ; 2) бокс, <b>вкладка</b> , коробка, ящик	відображає зображення
<b>Point</b>	[pɔɪnt]	пункт, <b>крапка</b>	New Point – нова крапка
<b>Preserve</b>	[prɪ'zɜ:v]	<b>зберігати</b> , оберігати	якщо модифікатор Preserve зазначений, значення елементів копіюються з існуючого масиву в новий
<b>Printdialog</b> 1) Print 2) Dialog	[prɪnt] [ˈdaɪəlɒɡ]	1) <b>печатка</b> , шрифт, друкування, друкувати 2) <b>діалог</b> , розмова	відображає діалогове вікно, що дозволяє користувачеві вибрати принтер і інші параметри печатки
<b>Printdocument</b> 1) Print 2) Document	[prɪnt] [ˈdɒkjument]	1) <b>печатка</b> , шрифт, друкування, друкувати 2) <b>документ</b> , текст, папір	визначає об'єкт, що відправляє вихідні дані на принтер
<b>Printform</b> 1) Print 2) Form	[prɪnt] [fɔ:m]	1) <b>печатка</b> , шрифт, друкування, друкувати 2) <b>форма</b>	дозволяє друкувати Windows Form, як простий звіт
<b>Printpreviewcontrol</b> 1) Print 2) Preview 3) Control	[prɪnt] [ˈpri:'vjʊ:] [ken'troul]	1) <b>печатка</b> , шрифт, друкування, друкувати 2) <b>попередній перегляд</b> , анонс 3) <b>контроль</b> , керування, боротьба	представляє тільки частину документа в режимі попереднього перегляду

## Продовження таблиці А.4

1	2	3	4
<b>Printpreviewdialog</b> 1) Print 2) Preview 3) Dialog	[ˈprint] [ˈpri:ˈvju:] [ˈdaɪəɒlg]	1) <b>печатка</b> , шрифт, друкування, друкувати 2) <b>попередній перегляд</b> , анонс 3) <b>діалог</b> , розмова	відображає діалогове вікно попереднього перегляду документа перед печаткою
<b>Private</b>	[ˈpraɪvɪt]	рядовий, солдат; <b>приватний</b> , особистий, відокремлений, секретний	Private Sub – приватний підрозділ
<b>Process</b>	[ˈprəʊses]	дія, <b>процес</b>	представляє доступ до локальних і вилучених процесів, забезпечуючи можливість запуску й зупинки локальних процесів
<b>Progressbar</b> 1) Progress 2) Bar	[ˈprəʊgres] [ba:]	1) <b>прогрес</b> , розвиток, успіхи, плин, поліпшення; 2) <b>риса</b> , важіль, смуга	відображає індикатор виконання операції
<b>Public</b>	[ˈpʌblɪk]	<i>ім.</i> : публіка, народ, громадськість; <i>прикм.</i> : публічний, загальнодоступний, <b>відкритий</b>	Public Class – відкритий клас
<b>Radiobutton</b> 1) Radio 2) Button	[ˈreɪdiəʊ] [bʌtn]	1) <b>радіо</b> , радіограма, радиувати; 2) <b>кнопка</b> , гудзик	дозволяє користувачу обрати певний параметр із групи в комбінації з іншими кнопками у вигляді прапорців
<b>Readalltext</b> 1) Read 2) All 3) Text	[ri:d] [ɔ:l] [tekst]	1) читати, вивчати, тлумачити, <b>читання</b> 2) усі, <b>увесь</b> , цілий 3) <b>текст</b> , текстовий файл, оригінал	Filesystem.Readalltext – ФайловаяСистема. ЧитатьВесьТекст
<b>Readline</b> 1) Read 2) Line	[ri:d] [laɪn]	1) читати, вивчати, тлумачити, <b>читання</b> 2) лінія, черга, <b>рядок</b>	Readline – метод виконує читання рядка символів із поточного потоку й повертає дані у вигляді рядка
<b>ReadOnly</b> 1) Read 2) Only	[ri:d] [ˈəʊnli]	1) читати, вивчати, тлумачити, <b>читання</b> 2) <b>тільки</b> , недавно	ReadOnly – задає значення, що вказує, чи може користувач змінювати комірки
<b>Refresh</b>	[rɪˈfreʃ]	освіжати, <b>оновляти</b>	Примусово створює умови, при яких елемент керування робить недоступною свою клієнтську область і негайно перемальовує себе й усі дочірні елементи
<b>Region</b>	[ˈri:dʒən]	<b>область</b> , регіон, сфера, район	End Region – кінець області

## Продовження таблиці А.4

1	2	3	4
<b>Reportviewer</b> 1) Report 2) Viewer	[ri'pɔ:t] [vju:ver]	1) доповідь, <b>звіт</b> , повідомлення, протокол 2) глядач, <b>програма перегляду</b>	відображає звіт
<b>Richtextbox</b> 1) Rich 2) Text 3) Box	[ritʃ] [tekst] [bɒks]	1) багатий, яскравий, густий, коштовний, <b>рясний</b> ; 2) <b>текст</b> , шрифт, тема; 3) коробка, ящик, бокс, <b>вкладка</b>	забезпечує додаткові здатності введення й редагування тексту
<b>Row</b>		<b>рядок</b> , ряд	Rows.Add – Рядка.Додати
<b>Rowcount</b> 1) Row 2) Count	[.kaunt]	1) <b>рядок</b> , ряд 2) уважати, поррахувати, рахунок, <b>кількість</b>	Повертає або задає число рядків, що відображаються в об'єкті DataGridView.
<b>Rowheadersvisible</b> 1) Row 2) Header 3) Visible	['hedə] ['vizəbl]	1) <b>рядок</b> , ряд 2) <b>заголовок</b> 3) <b>видимий</b> , явний	Rowheadersvisible – Указуємо, чи відображається стовпець, що містить заголовки рядків.
<b>Savefiledialog</b> 1) Save 2) File 3) Dialog	[seiv] [fail] ['daiəlg]	1) заощаджувати, <b>зберігати</b> , рятувати, зберегти 2) <b>файл</b> , досьє, архів, справа, зберігати, реєструвати 3) <b>діалог</b> , розмова	відображає діалогове вікно, що дозволяє користувачеві вибрати місце розташування для збереження файлу
<b>Selectedindexchanged</b> 1) Select 2) Index 3) Change	[sɪ'lekt] ['ɪndeks] [tʃeɪndʒ]	1) відбирати, <b>вибирати</b> 2) вносити в покажчик, постачати покажчиком, <b>індексувати</b> 3) <b>змінювати</b> , замінити, обмінювати	Selectedindexchanged – вибрані індексовані зміни
<b>Sender</b>	[ˈsendə]	<b>відправник</b> , передавач	sender As System.Object – відправник як системний об'єкт
<b>Serialport</b> 1) Serial 2) Port	[ˈsiəriəl] [pɔ:t]	1) серійний, <b>послідовний</b> , порядковий, періодичний 2) порт	представляє ресурс послідовного порту
<b>Servicecontroller</b> 1) Service 2) Controller	[ˈsɜ:vɪs] [kən'trɒlə]	1) <b>служба</b> , послуга, обслуговування, сервіс 2) <b>контролер</b> , регулятор, контролер, оператор	забезпечує можливість підключення до працюючої або встановленої служби Windows, направлення до неї запитів і керування ними
<b>Showdialog</b> 1) Show 2) Dialog	[ʃəʊ] ['daiəlg]	1) <b>показувати</b> , проявляти, демонструвати 2) <b>діалог</b> , розмова	Showdialog – показати діалог
<b>Size</b>	[saɪz]	2) <b>розмір</b> , розміром	Повертає або задає значення, що вказує, як визначається ширина стовпця
<b>Solidbrush</b> 1) Solid 2) Brush	['sɒlɪd] [brʌʃ]	1) солідний, ґрунтовний, міцний <b>суцільний</b> 2) малювати кистю, <b>кисть</b>	Solidbrush – Визначає кисть одного кольору

## Продовження таблиці А.4

1	2	3	4
<b>Statusstrip</b> 1) Status 2) Strip	[ˈsteitəs] [strip]	1) положення, статус, <b>стан</b> ; 2) <b>панель</b> , вузька смуга, невелика ділянка	показує користувачу інформацію про переглянутий об'єкт, компоненти об'єкта, і опис об'єкта
<b>Step</b>	[step]	<b>крок</b> , ступати, рухатися поетапно	For t = tn To tk Step th – для t = tn До tk крок th
<b>StreamReader</b> 1) Stream 2) Reader	[stri:m] [ˈri:də]	1) потік, <b>потоківий</b> , передавати потоком 2) читач, <b>засіб читання</b>	StreamReader – Реалізує Textreader, який зчитує символи з потоку байтів у певному кодуванні
<b>String</b>	[striŋ]	послідовність, ряд, ланцюг, <b>рядок</b>	As String = як рядок
<b>Strip</b>	[strip]	<b>панель</b> , вузька смуга, невелика ділянка	ToolStrip – панель інструментів
<b>Sub</b>	[sʌb]	<i>положення нижче чого-н., під чим-н., більш дрібний підрозділ.</i>	Private Sub – частка підрозділ
<b>System</b>	[ˈsistim]	<b>система</b> , організм	System.Object – системний об'єкт
<b>Text</b>	[tekst]	<b>текст</b> , текстовий файл, оригінал	Writealltext – писати весь текст
<b>Textbox</b> 1) Text 2) Box	[tekst] [bɒks]	1) <b>текст</b> , шрифт, тема; 2) коробка, ящик, бокс, <b>вкладка</b>	дозволяє користувачеві вводити текст і забезпечує редагування декількох рядків, маскування символів пароля
<b>Textreader</b> 1) Text 2) Reader	[tekst] [ˈri:də]	1) текст, текстовий файл, оригінал; 2) читач, засіб читання	Textreader – Представляє засіб читання, що дозволяє зчитувати послідовні набори знаків.
<b>Texturebrush</b> 1) Texture 2) Brush	[ˈtekstʃə] [brʌʃ]	1) структура, текстура, <b>фактура</b> 2) малювати кистю, <b>кисть</b>	Texturebrush є об'єктом Brush, що використовують зображення для заливання внутрішньої частини
<b>Then</b>		потім, <b>тоді</b> , потім	
<b>Timer</b>	[taimər]	<b>таймер</b> , годинник	компонент, що створює подія з інтервалами, певними користувачем
<b>To</b>		<b>до</b> , для, в, на, до	For t = tn To tk Step th – для t = tn До tk крок th
<b>Tool</b>	[tu:l]	<b>інструмент</b> , знаряддя, засіб, верстат	ToolStrip – панель інструментів
<b>ToolStrip</b> 1) Tool 2) Strip	[tu:l] [strip]	1) <b>інструмент</b> , знаряддя, засіб, верстат; 2) <b>панель</b> , вузька смуга, невелика ділянка	надає панелі інструментів і інші елементи інтерфейсу
<b>ToolStripcontainer</b> 1) Tool 2) Strip 3) Container	[tu:l] [strip] [kənˈteɪnə]	1) <b>інструмент</b> , знаряддя, засіб, верстат; 2) <b>панель</b> , вузька смуга, невелика ділянка; 3) посудина, <b>вмістище</b> , контейнер	створює по обом сторонам форми панелі

## Закінчення таблиці А.4

1	2	3	4
<b>Tooltip</b> 1) Tool 2) Tip	[tu:l] [tip]	<b>1) інструмент</b> , знаряддя, засіб, верстат; <b>2) кінець</b> , наконечник, нахилити, нахил, попереджати	відображає інформацію при наведенні покажчика миші на відповідний елемент керування
<b>Trackbar</b> 1) Track 2) Bar	[træk] [ba:]	<b>1) відслідковувати</b> , стежити <b>2) важіль, смуга</b>	
<b>Translatetransform</b> 1) Translate 2) Transform	[træns'leɪt] [træns'fɔ:m]	<b>1) переводити</b> , перетворювати <b>2) перетворювати</b> , трансформувати	Translatetransform – переутворювати
<b>Treeview</b> 1) Tree 2) View	[tri:] [vju:]	<b>1) дерево, древо</b> , вісь ; <b>2) огляд, вид</b> , думка, погляд, проекція;	показує ієрархічну колекцію позначених об'єктів, які можуть містити зображення
<b>True</b>	[tru:]	дійсний, вірний, <b>істина</b>	Visible = True – Видимий=Істина
<b>Val</b>		<b>скорочення: вартість, цінність</b>	Val – повертає числа, що містяться в рядку в якості числових значень
<b>Value</b>	['vælju:]	цінність, вартість, значення	
<b>Visible</b>	['vɪzəbl]	<b>видимий</b> , явний	Visible = False – Видимий=Неправда
<b>Webbrowser</b>	[web] [brauzer]	<b>веб-браузер</b>	дозволяє переглядати веб-сторінки всередині форми
<b>Width</b>	[wɪdθ]	<b>ширина</b>	Width Step 20 – ширина Крок 20
<b>With</b>	[wɪð]	з, за допомогою, через, при	With – виконує блок операторів із повторюваними посиланнями на один об'єкт
<b>Writealltext</b>	[raɪt] [ɔ:l] [tekst]	<b>1) писати</b> , написати <b>2) усі, увесь, цілий 3) текст</b> , текстовий файл, оригінал;	Writealltext – писати весь текст

**ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ**

1. Абрамян М. Э. Практикум по программированию на языках С# и VB.NET. / М. Э. Абрамян. 2-е изд. – Р-н/Д.: ЦВВР, 2007. – 514 с.
2. Балена Ф. Современная практика программирования на Microsoft Visual Basic и Visual C# / Ф. Балена, Дж. Димауро. – М.: Русская редакция, 2006. – 970 с.
3. Дубовцев А. В. Microsoft .NET в подлиннике / А. В. Дубовцев. – СПб.: БХВ-Петербург, 2004. – 850 с.
4. Нортроп Т. Основы разработки приложений на платформе Microsoft .NET Framework / Т. Нортроп, Ш. Уилдермьюс, Б. Райан. – М.: Русская редакция, 2007. – 730 с.
5. Рихтер Дж. Программирование на платформе Microsoft .NET Framework. Мастер-класс. / Дж. Рихтер. 3-е изд. – М.: Русская редакция, 2005. – 565 с.
6. Microsoft Corporation. Разработка Windows-приложений на Microsoft Visual Basic .NET и Microsoft Visual C# .NET. – М.: Русская редакция, 2003. – 645 с.
7. Зиборов В. В. Visual Basic 2010 на примерах / В. В. Зиборов. – СПб.: БХВ-Петербург, 2010. – 336 с.
8. Методичні вказівки до вивчення дисципліни «Економічна інформатика» (для студентів напряму підготовки 6.030601 «Менеджмент» усіх форм навчання) [Електронний ресурс] / укладачі М. Є. Корольов, Р. С. Кравченко. – Електрон. дані. – Горлівка: ДВНЗ «ДонНТУ» АДІ, 2012, 76 с. [МВ № 22/14-2012-11].
9. Методичні вказівки до навчальної комп'ютерної практики. Частина 1. «Електронні таблиці MS Excel» (для студентів напрямів підготовки 6.030601 «Менеджмент», 6.070101 «Транспортні технології (автомобільний транспорт)» усіх форм навчання) [Електронний ресурс] / укладачі: М. Є. Корольов, Р. С. Кравченко. – Електрон. дані. – Горлівка: ДВНЗ «ДонНТУ» АДІ, 2012, 77 с. [МВ № 22/16-2012-15].

ЕЛЕКТРОННЕ НАВЧАЛЬНО-МЕТОДИЧНЕ ВИДАННЯ

**Корольов** Марк Євгенович  
**Кравченко** Роман Сергійович

**МЕТОДИЧНІ ВКАЗІВКИ  
ДО ПРАКТИКУМУ З РОЗДІЛУ «ОСНОВИ СИСТЕМ  
АВТОМАТИЗАЦІЇ ОБРОБКИ ЕКОНОМІЧНОЇ ІНФОРМАЦІЇ»  
З ДИСЦИПЛІНИ «ЕКОНОМІЧНА ІНФОРМАТИКА» (ДЛЯ  
СТУДЕНТІВ НАПРЯМУ ПІДГОТОВКИ 6.030601  
«МЕНЕДЖМЕНТ» УСІХ ФОРМ НАВЧАННЯ)**

Підписано до випуску 8.10.2013 р. Гарнітура Times New.  
Умов. друк. арк. 6,43. Зам. № 269.

---

Державний вищий навчальний заклад  
«Донецький національний технічний університет»  
Автомобільно-дорожній інститут  
84646, м. Горлівка, вул. Кірова, 51  
E-mail: druknf@rambler.ru

Редакційно-видавничий відділ

Свідоцтво про внесення до Державного реєстру видавців, виготовників і розповсюджувачів  
видавничої продукції ДК № 2982 від 21.09.2007 р.