

МІНІСТЕРСТВО АГРАРНОЇ ПОЛІТИКИ УКРАЇНИ  
Вінницький державний аграрний університет

Кафедра економічної кібернетики  
та інформатики

## **МЕТОДИЧНІ ВКАЗІВКИ**

для виконання лабораторних робіт з дисципліни  
„Інформатика та комп'ютерна техніка”  
для студентів стаціонарної форми навчання  
спеціальностей економічного спрямування

*(Visual Basic)*

Вінниця – 2004

УДК 004:371.214.144

Берник О.В., Паламарчук Є.А., Бісікало О.В., Станчук К.І., Бурденюк І.І.

Методичні вказівки для виконання лабораторних робіт з дисципліни „Інформатика та комп'ютерна техніка” для студентів стаціонарної форми навчання спеціальностей економічного спрямування (*Visual Basic*). – Вінниця: ВДАУ, 2004. – 60 с.

У методичних вказівках з дисципліни „Інформатика та комп'ютерна техніка” містяться теоретичні відомості та завдання для виконання циклу лабораторних робіт по вивченню мови візуального програмування *Visual Basic*.

Автори:

Рецензенти:

Схвалено на засіданні кафедри економічної кібернетики та інформатики  
Вінницького державного аграрного університету, протокол № \_\_\_ від \_\_\_  
\_\_\_\_\_ 2004 року

Схвалено науково-методичною радою Вінницького державного аграрного  
університету, протокол № \_\_\_ від \_\_\_ \_\_\_\_\_ 2004 року

## Лабораторна робота № 1. (4 год.)

**Тема:** Знайомство з *Visual Basic*. Елементи інтегрованого середовища. Елементи управління. Робота з проектом.

**Мета:** Ознайомитись з особливостями інтегрованого середовища розробки *Visual Basic*, елементами управління, навчитись створювати найпростіші програми.

### Теоретичні відомості

*Visual Basic* є потужною системою програмування, і забезпечує можливість зручного та ефективного створення складних програм або додатків, які працюють в ОС *Microsoft Windows*.

Процес побудови простого додатку полягає у розміщенні на формі елементів управління, зміні деяких властивостей та написанні деякого тексту програми, що визначається змістом і призначенням додатку.

Робоче середовище у *Visual Basic* називається інтегрованим середовищем розробки (*integrated development environment*) або IDE, тому що об'єднує велику кількість різних функцій, таких як розробка (*design*), редагування (*editing*), компіляцію (*compiling*) і налагоджування (*debugging*), всередині спільного середовища, на відміну від попередніх у часі засобів розробки програмних продуктів, де кожна окрема функція виконувалась іншою програмою.

Запустити *Visual Basic* можна різними способами, наприклад:

- 1) Пуск – Программы – *Microsoft Visual Basic*;
  - 2) знайти на Робочому столі відповідний ярлик та клацнути на ньому мишкою.
- Після завантаження *Visual Basic* на екрані з'явиться діалогове вікно (рис.1).

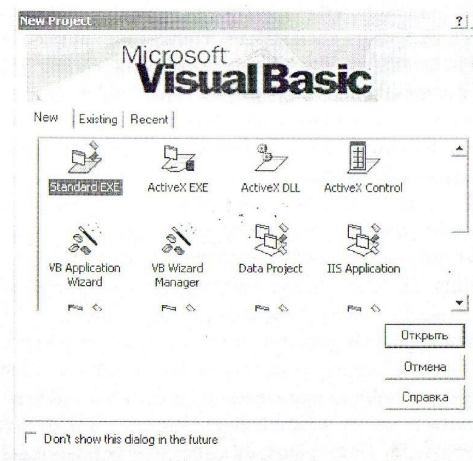


Рис.1.

У цьому вікні потрібно натиснути кнопку Открыть. Далі на екрані з'являється вікно, що показано на рис.2.

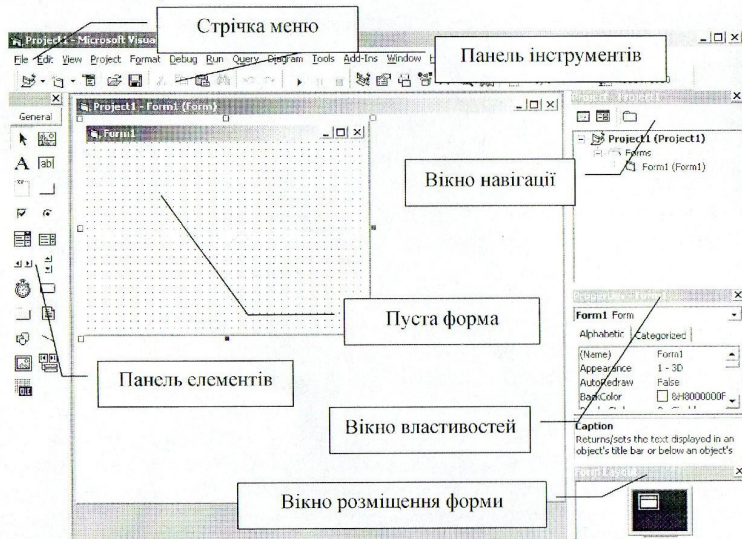


Рис. 2.

**Стрічка меню** містить команди для роботи з *Visual Basic*.

**Панель інструментів** забезпечує доступ до команд, що використовуються найчастіше. Якщо цієї панелі немає, то її можна відкрити обравши пункти меню *View – Toolbars – Standard* (в перекл. з англ. Вид – Панелі інструментів – Стандартна). Крім цієї панелі інструментів можна ще відкрити такі як *Debug* (Налагоджування), *Edit* (Редагування), *Form Editor* (Редактор форми).

**Панель елементів** призначена для вибору та розміщення на формі різних елементів управління. Якщо ця панель закрита, то відкрити її можна через пункти меню *View – Toolbox* (Вид – Елементи).

**Вікно навігації** служить для переходу між об'єктами проекту: списком форм і модулів. Це вікно можна відкрити через пункти меню *View – Project Explorer*.

**Вікно властивостей** містить перелік встановлених властивостей вибраної форми чи елемента управління. Це вікно відкривається через пункти меню *View – Properties Window*.

**Вікно розміщення форми** показує в якому місті екрану з'являтиметься форма після запуску програми. Перетягуючи це вікно мишкою можна вибрати бажане розміщення. Вікно відкривається через пункти меню *View – Form Layout Window*.

Створення власного додатку у *Visual Basic* можна умовно поділити на **3 етапи**:

- 1) створення інтерфейсу (вибираються елементи управління, що будуть міститися у програмі, створюється меню та панелі інструментів, тощо);
- 2) зміна властивостей елементів інтерфейсу;

3) написання тексту програми.

Розглянемо **перший етап**. Фундаментом для створення інтерфейсу будь-якого додатку є форма (рис.2).

На формі можна розмістити елементи управління (рис.3). Для цього є два способи:

- 1) клацнути двічі лівою кнопкою мишки на потрібному елементі панелі елементів (*General*);
- 2) натиснути потрібний значок на панелі елементів лівою кнопкою мишки і на формі розтягнути мишкою прямокутник (місце розміщення елемента та його розмір).

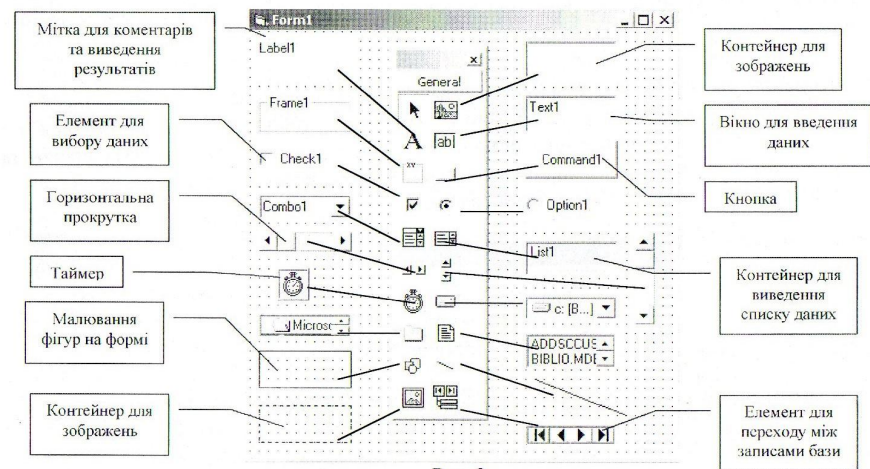


Рис.3.

**Другий етап** полягає у зміні властивостей того чи іншого елемента. Для цього його потрібно виділити (натиснути один раз лівою кнопкою мишки) та перейти у вікно властивостей.

Так само як і елементи, форма теж має властивості. Нижче подано перелік деяких з них (для ознайомлення).

**BackColor** – кольори фону для форми.

**BorderStyle** – особливості границь форми (границі можуть бути фіксованими або масштабними).

**Caption** – текст, який виводиться в заголовку форми.

**ControlBox** – наявність у заголовку форми кнопки системного меню (*True* or *False*).

**ForeColor** – кольори тексту.

**Height** – висота форми.

**Icon** – значок, який виводиться на панель задач при згортанні форми в режимі виконання.

**Left** – визначає відстань від форми до лівого краю екрану.

**Name** – вказує ім'я, по якому *Visual Basic* визначає об'єкт під час виконання програми.

**Width** – визначає ширину форми в твіпах.

**WindowState** – визначає стан форми під час запуску: 0 – *Normal* – форма відкривається у нормальному вигляді, 1 – *Minimized* – форма відкривається в згорнутому вигляді, 2 – *Maximized* – форма розгортається на весь екран.

Наприклад, для того щоб у вікні на рис.3 в заголовку замість *Form1* з'явився інший напис, потрібно виділити форму та у стрічці *Caption* ввести нову назву.

Якщо на формі розмістити всі потрібні елементи та задати для них потрібні властивості, то отримаємо не кінцевий програмний продукт, а лише його візуальне уявлення. Для того, щоб програма працювала необхідно пов'язати елементи з певними подіями.

### Третій етап.

Переглянути події, з якими пов'язані об'єкти можна, якщо натиснути на об'єкті двічі лівою кнопкою мишки. На рис.4 зображено вікно, яке відкривається при натисненні двічі лівої кнопки мишки на об'єкті Форма. Запис *Form* (зліва) означає, що об'єкт Форма пов'язується з якоюсь подією, а саме, в даному випадку подією *Load* (справа), що в перекладі означає завантаження.

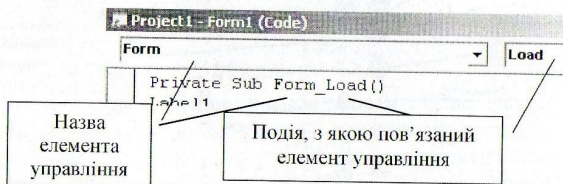


Рис.4.

Далі, необхідно вказати дії, які виконуватиме програма після завантаження форми. Перелік цих дій (або команд) називається **процедурою**.

Процедури поділяються на 2 категорії:

- процедури-підпрограм (підпрограми–*Sub*)
- процедури-функції(функції–*Function*).

Поки що будемо розглядати процедури-підпрограми. Її синтаксис наступний:

```
Private Sub [Назва процедури] ()
```

```
Команда 1
```

```
...
```

```
Команда n
```

```
End Sub
```

Запис *Form Load()*, означає, що після завантаження форми необхідно виконати всі команди до слів *End Sub*.

У програмі ми можемо змінювати властивості елементів (їх зовнішній вигляд, розташування, тощо). Крім того, для кожного елементу існує ряд **методів**, що дозволяє виконувати з ним якісь дії. Переглянути властивості чи методи, які має той чи інший елемент можна якщо в тілі програми набрати назву елемента та поставити крапку (рис.5).

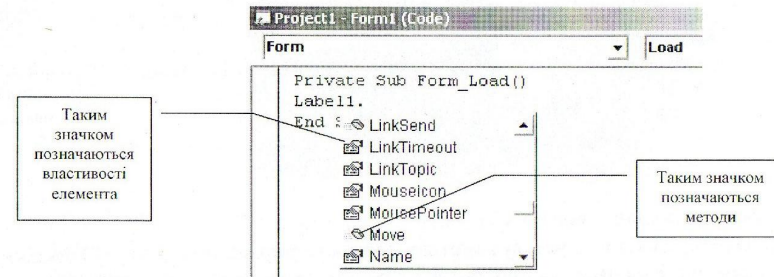




Рис.5.

Для того, щоб програма виконалась потрібно натиснути кнопку на панелі інструментів , відповідно наступні кнопки  призначені для паузи та зупинки програми.

Після створення проекту, його потрібно зберегти.

Проект зазвичай складається з декількох файлів:

- одного файлу проекту, в якому зберігаються всі компоненти проекту, файл має розширення *\*.vbp*;
- одного файлу для кожної форми (*\*.frm*);
- одного війкового файлу даних для кожної форми з даними про властивості елементів управління на формі (*\*.frx*). Цей файл генерується автоматично;
- один файл для кожного модуля класу (*\*.cls*). Цей файл не обов'язковий;
- один файл для кожного стандартного модуля (*\*.bas*). Не обов'язковий;
- один чи більше файлів, що містять Active-X елементи (*\*.ocx*). Не обов'язковий;
- єдиний ресурсний файл (*\*.res*), не обов'язковий.

Для того, щоб зберегти файл необхідно вибрати пункт головного меню *File* (Файл) – *Save project* (Зберегти проект). У діалоговому вікні необхідно вказати папку для збереження проекту (нагадаємо, що зберігати файли можна лише на власному диску, диск X:\) та ім'я файлу. Спочатку зберігається файл форми (*\*.frm*), а потім файл програми (проекту) (*\*.vbp*), на всі послідовні запити слід відповідати *No* (Ні).

Для того, щоб отримати кінцевий виконуваний файл, який буде запускатися з *Windows* потрібно вибрати пункти меню *File-Make \*.exe* (Замість зірочки буде вказане ім'я, під яким ви зберегли проект).

Відкрити збережений проект можна шляхом вибору та запуску відповідного файлу з розширеннями *\*.frm* або *\*.vbp*.

Якщо на екрані з'явилося вікно *Visual Basic*, але в ньому немає Вашого проекту, то зверніть увагу на вікно навігації, якщо його вид такий, як на рис.6, то слід натиснути на „+” у цьому вікні, а потім двічі на стрічку, що з'явиться.

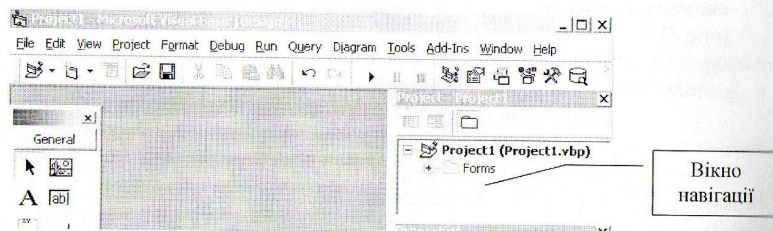


Рис.6

Розглянемо приклад створення проекту.

Вам пропонується програма, яка містить елементи управління: мітку і 3 кнопки. У мітці відображено інформацію про автора проекту, при натисканні першої кнопки змінюється колір фону форми, по натисканні другої – з'являється деяке повідомлення, натиснувши третю кнопку можна вийти з програми.

Розмістіть на формі елементи управління мітку та 3 кнопки.

Змініть для них властивості:

Назва об'єкту та властивість	Значення
<i>Form1.Caption</i>	Мій перший проект
<i>Form1.BackColor</i>	Перейдіть на закладку <i>Palette</i> та оберіть колір, який вам сподобається
<i>Label1.Caption</i>	Цю роботу виконав студент групи 16 ОА (вказуйте власну групу та прізвище)
<i>Label1.Alignment</i>	2-Center (цією властивістю визначається положення тексту в комірці: з правого краю, по центру та з лівого краю)
<i>Label1.BorderStyle</i>	1-Fixed Single (цією властивістю визначається тим границь мітки, може бути простий та об'ємний)
<i>Label1.BackColor</i>	Перейдіть на закладку <i>Palette</i> та оберіть колір, який вам сподобається
<i>Label1.Font</i>	Цією властивістю обирається тип шрифту та розмір, оберіть <i>Times New Roman</i> , жирний, 14 пт.
<i>Command1.Caption</i>	Змінити колір фону форми
<i>Command2.Caption</i>	Прочитати повідомлення
<i>Command3.Caption</i>	Вийти з програми

Форма набуде наступного виду (рис.7)

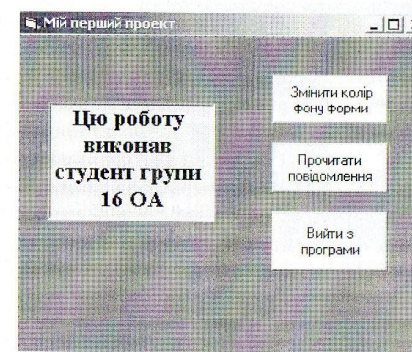


Рис.7.

Переходимо до третього етапу роботи над проектом – написання програми.

Натисніть двічі лівою кнопкою мишки на кнопку „Змінити колір фону форми”.

У вікні, що з'явилося наберіть текст програми (рис.8). Для того, щоб не писати кожний раз назву процедури, можна натискати двічі на відповідну кнопку (другу „Прочитати повідомлення” та третю „Вийти з програми”).

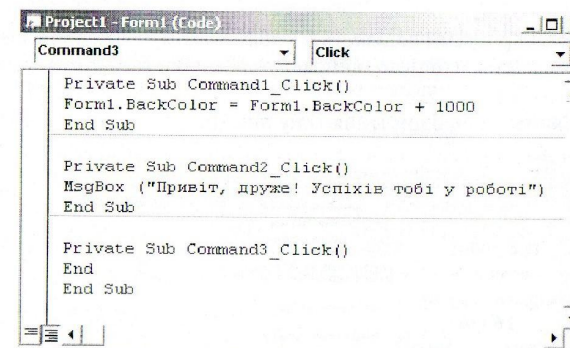


Рис.8.

Запустіть програму на виконання кнопкою (Start).

Під час запуску програми може з'явитися повідомлення про помилку, наприклад, таке, як показане на рис.9.

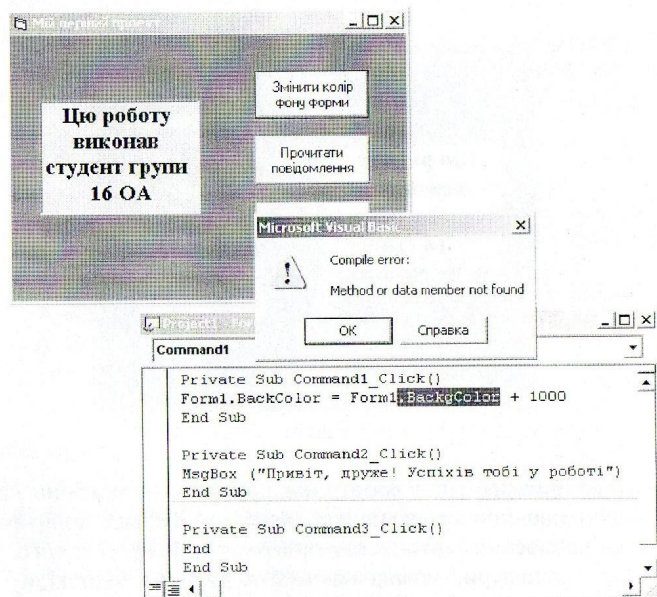


Рис.9.

Одразу в тексті програми виділяється слово у якому може бути помилка. Тоді виконання програми слід зупинити, виправити помилку та запустити на виконання знову.

Або може з'явиться повідомлення типу рис.10.

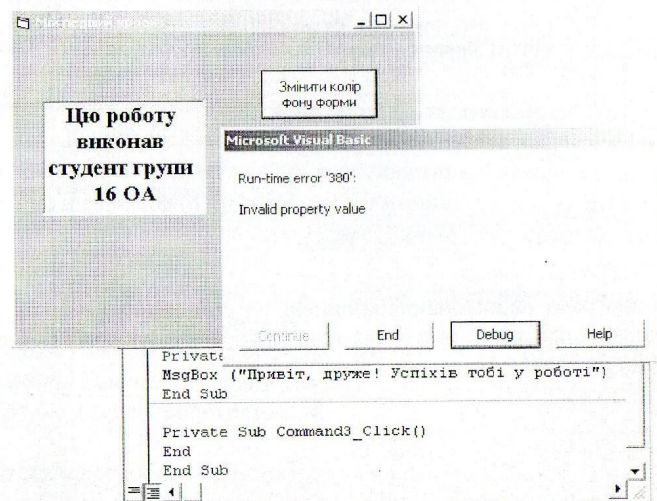


Рис.10

Воно також означає наявність якоїсь помилки. У цьому випадку потр спочатку натиснути кнопку *Debug* (налагоджування), у тексті програми жом кольором виділиться стрічка, в якій може бути помилка. Потім програму зупинити, виправити помилку та запустити на виконання знову.

Одними із найпоширеніших помилок є:

- неправильне написання команд;
- використання літер українського алфавіту при написанні тексту прог (текст програми пишеться лише англійськими буквами, українськ можна писати лише текст у лапках та коментарі);
- невідповідність тексту програми елементам, що розміщені на формі та

### Порядок виконання роботи

1. Ознайомтесь з теоретичною частиною.
2. Запустіть *Visual Basic* та дослідіть меню, панелі інструментів та вікна.
3. Відтворіть проект поданий у прикладі .
4. Збережіть проект у папці X:\Lab1\.
5. Створіть \*.exe файл вашого проекту.
6. Закрийте даний проект.
7. Створіть новий проект. Цей проект демонструє використання елеме *OptionButton* та *Shape*.

Нижче подано інструкцію до виконання цього проекту.

Розмістіть на формі три елементи *OptionButton* та три елементи *Shape*.

Змініть для них властивості так, як показано в таблиці.

Назва об'єкту та властивість	Значення
<i>Form1.Caption</i>	Світлофор
<i>Form1.BackColor</i>	Будь-який колір
<i>Option1.Value</i>	<i>True</i>
<i>Option1.BackColor</i>	Колір фону форми
<i>Option1.Caption</i>	Червоний
<i>Option2.BackColor</i>	Колір фону форми
<i>Option2.Caption</i>	Жовтий
<i>Option3.BackColor</i>	Колір фону форми
<i>Option3.Caption</i>	Зелений
<i>Shape1.Shape</i>	3-Circle
<i>Shape1.BackColor</i>	Червоний
<i>Shape1.BackStyle</i>	1- <i>Opaque</i>
<i>Shape2.Shape</i>	3-Circle
<i>Shape2.BackColor</i>	Сірий
<i>Shape2.BackStyle</i>	1- <i>Opaque</i>
<i>Shape3.Shape</i>	3-Circle
<i>Shape3.BackColor</i>	Сірий
<i>Shape3.BackStyle</i>	1- <i>Opaque</i>

Форма набуде наступного виду (рис.11).

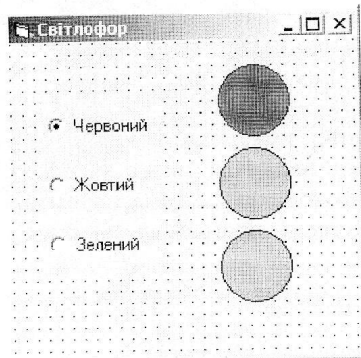


Рис.11.

Натисніть двічі лівою кнопкою мишки на елементи *Option1*, потім *Option2*, потім *Option3*.

Текст програми матиме наступний вид (рис.12):

```
Project1 - Form1 (code)
Option1 Click
Private Sub Option1_Click()
Shape1.BackColor = &HFF&
Shape2.BackColor = &H808080
Shape3.BackColor = &H808080
End Sub

Private Sub Option2_Click()
Shape1.BackColor = &H808080
Shape2.BackColor = &H80FFFF
Shape3.BackColor = &H808080
End Sub

Private Sub Option3_Click()
Shape1.BackColor = &H808080
Shape2.BackColor = &H808080
Shape3.BackColor = &HFF00&
End Sub
```

Рис.12

Виконайте програму.

- Збережіть файли проекту (не забудьте змінити назву форми *Form1* на *Form2*, та проекту *Project1* на *Project2*).
- Створіть виконуваний файл цього проекту.

Звіт повинен містити тему, мету, короткий опис виконаної роботи та виснов. Для захисту роботи необхідно продемонструвати викладачу на ЕОМ і проекти, показати звіт та дати відповіді на контрольні запитання.

### Контрольні запитання

- Що таке інтегроване середовище розробки?
- Як запустити *Visual Basic*?
- Які елементи вікна робочого середовища *Visual Basic* ви знаєте?
- Як відкрити панель елементів?
- Як відкрити вікно навігації?
- Як відкрити вікно властивостей?
- Які є етапи роботи над проектом, коротко охарактеризуйте кожен етап.
- Які елементи управління можна розмістити на формі, як це зробити?
- Як змінити властивість елемента?
- Що таке події?
- Що таке методи?
- Який синтаксис процедури-підпрограми?
- Як називається властивість за допомогою якої можна змінити колір фону форми чи мітки?
- Як називається властивість за допомогою якої можна змінити заголовок форми чи текст у мітці?
- Які обов'язкові файли містить проект *Visual Basic*?
- Як зберегти проект?
- Як запустити проект на виконання?
- Як відкрити збережений проект?
- Як зупинити виконання проекту?
- Як зробити виконуваний \*.exe файл?

## Лабораторна робота №2. (4 год.)

**Тема:** Типи даних. Математичні оператори та функції *Visual Basic*. Функції перетворення типів. Складання лінійних програм.

**Мета:** Розглянути основні типи даних, математичні оператори та функції, що використовуються у *Visual Basic*. Навчитись складати найпростіші лінійні програми.

### Теоретичні відомості

Дані чи змінні, якими оперують у програмах *Visual Basic* в переважній більшості випадків потребують описання. Тому що, в залежності від типу змінної, для її зберігання у пам'яті виділяється певна кількість місця.

Наприклад, нехай  $S = x + y$ , перед тим, як безпосередньо записати цю дію у програму необхідно продумати, яких значень зможуть набувати числа  $x$  та  $y$ , а також якого значення набуде результат. Якщо  $x$  та  $y$  цілі числа, то результат теж буде цілим числом ( $x = 2, y = 2, S = 2 + 2 = 4$ ), якщо ж один із доданків буде дробовим, то сума теж буде дробовим числом ( $x = 2, y = 2.5, S = 2 + 2.5 = 4.5$ ), якщо змінні будуть буквенною стрічкою, то результат теж буде мати відповідний тип ( $x = 'abc', y = 'edf', S = 'abc'+'edf' = 'abcdf'$ ) і т.д.

Дані можуть бути різних типів:

**Числові** – тобто змінні можуть бути лише числами. Це типи *Integer, Long, Single, Double* і *Currency*. Нижче, у таблиці подано опис даних цих типів:

Назва	Опис
<i>Integer</i>	Цілі числа з діапазону від -32 768 до +32 676
<i>Long</i>	Цілі числа в діапазоні від -2 147 483 648 до +2 147 483 647
<i>Single</i>	Дробові чи цілі числа на проміжку від -3.402823E+38 до +3.402823E+38
<i>Double</i>	Дробові чи цілі числа в діапазоні від -1.79769313486232D+308 до +1.79769313486232D+308
<i>Currency</i>	Значення грошей від -\$922 337 203 685 477.5808 до +\$922 337 203 685 477.5808

До числового також можна віднести тип *Byte*, цим типом описуються змінні, що набувають додатних цілих значень від 0 до 255.

**Рядкові** (*String*) – змінні можуть бути стрічками, довжиною від 0 до 654000 алфавітно-цифрових символів.

**Логічного** (*Boolean*) – змінні можуть набувати двох значень *True* чи *False*.

**Тип дати** (*Date*) – значення змінних можуть бути в діапазоні від 1 січня 100 року до 31 грудні 9999 року.

У випадку, коли тип результату невідомий, змінну можна описати типом *Variant*, але в цьому випадку для зберігання кожної змінної виділяється максимальна кількість пам'яті, що в кінцевому результаті буде уповільнювати виконання програми.

Описання даних проводиться у підрозділі програми, що має таку конструкцію:

**Dim** (перелік змінних) **As** (тип змінних).

Для прикладів, наведених вище матимемо наступні стрічки опису змінних:

1) припустимо, що в програмі  $x=3, y=8$ , потрібно обчислити за формулою  $S=x$   $x$  і  $y$  – цілі числа, значить і результат буде дорівнювати 11, тобто є ціл числом. Такі змінні опишуться наступним чином:

*Dim x, y, S As Integer*

2) припустимо, що  $x=5, y=6.28, S=xy$ . У цьому випадку цілим числом буде тільки  $x, y$  і  $S$  будуть дробовими числами, тоді стрічки опису матимуть вид:

*Dim x As Integer*

*Dim y, S As Single*

3) припустимо, що  $x="123", y="абв", S=x+y$ , тоді  $S$  буде дорівнювати стрічку „123абв”, і тоді самі змінні і результат будуть віднесені до рядкового типу:

*Dim x, y, S As String*

Результат обчислень та змінні, що використовуються у виразі повинні бути одного типу. Для того, щоб перетворити один тип даних у інший у *Visual Basic* необхідно використати функції. У цій роботі ми розглянемо дві з них.

Функція (аргумент)	Дія
<i>Str(N)</i>	Перетворює число $N$ в текст
<i>Val(S)</i>	Перетворює текст (стрічку $S$ ) в число, але текст повинен містити тільки цифри та одну десяткову крапку, інакше перетворення не можливе. Як <i>Visual Basic</i> не може перетворити текст на число, то функція поверне значення 0

Приклад використання функції перетворення даних:

$x = Str(123)$

$y = Str(456)$

$S = x + y = 123456$

При створенні програм у середовищі *Visual Basic* можна використовувати такі арифметичні операції:

“+” - додавання;

“-” - віднімання;

“\*” - множення;

“/” - ділення;

“^” - піднесення до степеня;

та операції порівняння: “=”, “<”, “>”, “<>”, “>=”, “<=”.

Крім того, того *Visual Basic* містить набір математичних функцій. Нижче таблиці подано назви та короткий опис основних функцій.



Функція (аргумент)	Дія
$Abs(N)$	Повертає абсолютне значення $N$ .
$Atn(N)$	Повертає арктангенс $N$ як кут в радіанах.
$Cos(N)$	Косинус кута $N$ , $N$ вимірюється в радіанах.
$Exp(N)$	Повертає константу $e$ , піднесену до степеня $N$ ( $e$ – основа натурального логарифму, дорівнює приблизно 2.718282).
$Fix(N)$	Повертає цілу частину $N$ . $Fix$ не округляє число, а відкидає дробову частину. Якщо $N$ – від’ємне число, то $Fix$ повертає найближче від’ємне ціле число більше, ніж $N$ .
$Int(N)$	Повертає цілу частину $N$ . $Int$ не округляє число, а відкидає дробову частину. Якщо $N$ – від’ємне число, то $Int$ повертає найближче від’ємне ціле число менше, ніж $N$ .
$Log(N)$	Повертає натуральний логарифм $N$ .
$Rnd(N)$	Повертає випадкове число; аргумент є необов’язковим.
$Sgn(N)$	Повертає знак числа: -1, якщо $N$ – від’ємне; 1 – якщо $N$ додатне; 0 – якщо $N$ дорівнює 0.
$Sin(N)$	Повертає синус кута; $N$ – вимірюється в радіанах.
$Sqr(N)$	Повертає корінь квадратний із $N$ . Аргумент повинен бути додатнім числом.
$Tan(N)$	Повертає тангенс кута; $N$ – кут в радіанах.

Приклад. Скласти програму обчислення  $y$  за формулою, якщо відомі  $x, a, b, c$  (значення цих змінних повинно вводити з клавіатури).

Розмістіть на формі елементи управління так, як це показано на рис. нижче.

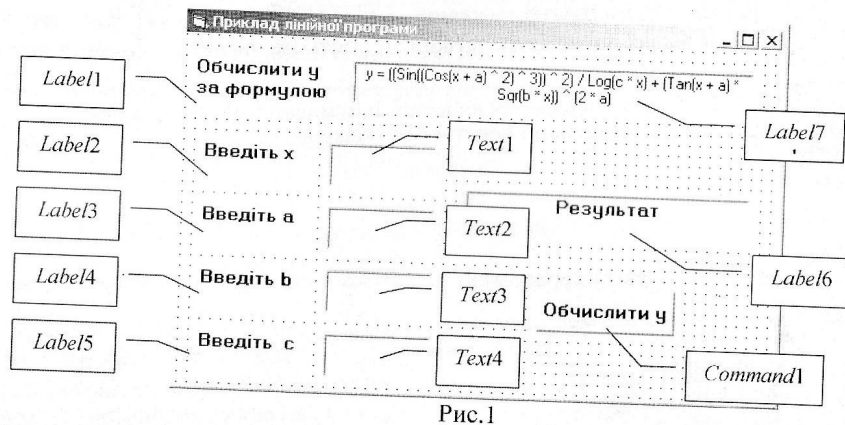


Рис.1

Елементи управління та їх змінені властивості вказані у наступній таблиці.

Назва	Властивості
<code>Form1.Caption</code>	Приклад лінійної програми
<code>Label1.Caption</code>	Обчислити $y$ за формулою
<code>Label2.Caption</code>	Введіть $x$
<code>Label3.Caption</code>	Введіть $a$
<code>Label4.Caption</code>	Введіть $b$
<code>Label5.Caption</code>	Введіть $c$
<code>Text1.Text</code>	
<code>Text2.Text</code>	
<code>Text3.Text</code>	
<code>Text4.Text</code>	
<code>Label6.Caption</code>	Результат
<code>Label6.BorderStyle</code>	1-Fixed Single
<code>Label7.Caption</code>	$y = ((\sin((\cos(x+a)^2)^3))^2) / \log(c*x) + (\tan(x+a) * \text{Sqr}(b*x))^{2*a}$
<code>Label7.BorderStyle</code>	1-Fixed Single
<code>Label7.Alignment</code>	2-Center
<code>Label7.BackColor</code>	Білий

Якщо натиснути мишкою кнопку “Обчислити  $y$ ”, то перейдемо у вікно прогн Програма матиме вид (рис.2), а результат роботи програми показано на рис.3.

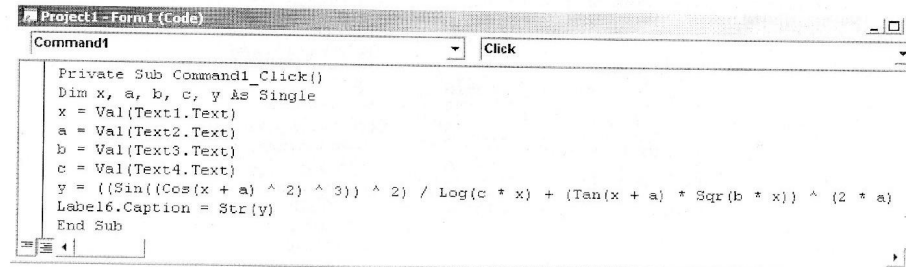


Рис.2

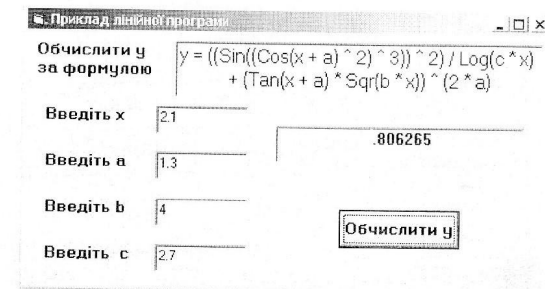


Рис.3

### Порядок виконання роботи

- Ознайомитись з теоретичним матеріалом.
- Відтворити поданий у прикладі проект.
- Скласти лінійну програму для обчислення значення виразу за варіантом (варіант вказує викладач),  $x$  та  $a$ ,  $b$ ,  $c$  повинні вводитись з клавіатури та набувати дробових значень:

Варіант	Вираз
1	$y = \frac{\operatorname{tg} \sqrt{(b+x)^3}}{x^3(a+x)^4} + c^{-9b} + \frac{\sin x}{\cos(\lg 2x)}$
2	$y = \frac{a + \sqrt{\lg(\sin b + \cos c)}}{\sqrt[4]{(1+x^2)}} + e^{\sin x^5}$
3	$y = \sin^4 x + a \cos^3 x + \left( \frac{\sqrt{\lg(\operatorname{arctg} b)}}{\sin \sqrt{c}} \right)^c$
4	$y = \frac{ a - \operatorname{tg} x }{1 + \operatorname{tg} x} + \sin \left( 2 \cos \frac{b^x}{\operatorname{tg} c^2} \right)$
5	$y = \frac{ax}{e^x \sin x} + \frac{bc -  \sin x^{5c} }{\cos( \lg(c-a) )}$
6	$y = ax^e \ln x + \frac{e^{\cos \operatorname{tg} b }}{\operatorname{tg}(\cos^5 c) + x^2 + \lg x}$
7	$y = \frac{a \ln^2 x}{\sqrt{x}} + \frac{b}{\cos^3 c} + \frac{x^5}{\operatorname{tg} a}$
8	$y = \frac{a}{\ln(\operatorname{tg} 2x)} + \sqrt{\frac{3b + c^e - a}{\cos(\sin^3 x)}}$
9	$y = \frac{\arccos x}{ax^2} \cdot \sqrt[3]{\frac{\ln bc }{\operatorname{tg}(\sin bx)}}$
10	$y = \operatorname{arctg} \frac{ax}{2} + \sqrt[5]{\frac{e^{bc} \cdot \sin \cos c }{ a^{-2bc} }}$
11	$y = \frac{b^{-0.5x} + \sqrt{ \sin x + \cos x^2 }}{\sin(\cos^2 x) + a \cdot  \cos(\sin cx^2)  + c}$
12	$y = \frac{\frac{1}{x} + \cos \left  \frac{x^3}{\sqrt{x^2 + 1}} \right  - e^{\cos \sin x }}{a^{c-bx} + e^{x+1}}$

13	$y = \frac{\ln ax  \cdot \operatorname{tg} \sqrt{ x+a } + x^{0.3b}}{a \cdot \operatorname{tg} x \cdot e^2 + c \cdot e^{a-x}}$
14	$y = \frac{e^{-\ln x  + \cos x} + e^{a \cdot \ln x+1  - \cos x}}{e^{\frac{1}{a-x}} + b \cdot e^{\frac{c}{x+a}}}$
15	$y = \frac{(a \cdot \sin bx +  \cos x - c )^{1.3}}{\sin cx + \sqrt{\ln(x^2 + 1)}}$

4. Скласти програму для конвертації валют (за варіантом). Передбачити можливість зміни курсу. Кількість коштів та курс повинні вводитись з клавіатури, як результат відображається конвертована сума.

На формі слід розмістити:

- елементи для введення суми, що підлягає конвертації та курсу, це можуть бути елементи *Text1* та *Text2*;
- елементи для коментарів про те, що саме потрібно вводити, це можуть бути *Label1* та *Label2*;
- елемент для виведення результату, *Label3*;
- кнопку, по натисненні якої буде здійснюватися розрахунок конвертованої суми. Властивості кожного з цих елементів змінюють за власними вподобаннями.

Варіант	Завдання
1	Гривні в євро
2	Долари США в євро
3	Євро у англійські фунти
4	Долари США в гривні
5	Гривні у єни
6	Англійські фунти в єни
7	Гривні у російські рублі
8	Російські рублі в долари
9	Єни в євро
10	Гривні в долари США
11	Долари США в англійські фунти
12	Російські рублі в євро
13	Російські рублі в англійські фунти
14	Англійські фунти в гривні
15	Єни в гривні

Звіт з лабораторної повинен містити:

- тему, мету;
- блок-схему алгоритмів до індивідуальних завдань;
- схематичний вид розташування об'єктів на формі;

- програми;
- висновки.

### Контрольні запитання

1. Які існують типи даних у *Visual Basic*, їх характеристики?
2. Яким чином описуються змінні у програмі?
3. Якщо змінна є текстом, то як вона буде описана у *Visual Basic*?
4. До якого типу відносяться змінні, що використовуються у грошових розрахунках?
5. Які математичні оператори використовуються у *Visual Basic*?
6. Назвіть основні математичні функції у *Visual Basic* та вкажіть їх призначення?
7. Для чого використовуються функції перетворення типів? Наведіть приклади.
8. Якою властивістю змінюється вид границь мітки?
9. Яке повідомлення з'явиться на екрані і чому, якщо спробувати обчислити у, але при цьому не ввести значення змінних  $x, a, b, c$ ?
10. Якою є послідовність складання лінійних програм?

### Лабораторна робота №3. (4 год.)

**Тема:** Програми з розгалуженням. Оператори *If, Select Case*. Налаштування програм Обробка помилок.

**Мета:** Розглянути умовні оператори для прийняття рішень, що використовуються *Visual Basic*. Навчитись складати та налагоджувати найпростіші розгалужені програм

### Теоретичні відомості

Оператори структур розгалуження *If... Then... Else* (в перекл. з англ. Якщо ... То Інакше)

Оператори *If... Then... Else* є найпростішою і найпоширенішою формою перевірки умов у *Visual Basic*. Його скорочена форма має наступний синтаксис:

*If* (Умова) *Then*

(Вираз1)

(Вираз2)

...

(Вираз ...)

*End If*

Умова записується так: (Вираз1) Оператор порівняння (Вираз2). Оператор порівняння може бути відповідне слово або знак:

Оператор, знак	Дія
<i>AND</i>	Якщо обидві умови мають значення <i>True</i> (в перекл. з англ. Істина), то результат перевірки умов має значення <i>True</i>
<i>OR</i>	Якщо одна з умов має значення <i>True</i> , то результат перевірки умов має значення <i>True</i>
<i>NOT</i>	Якщо умова має значення <i>True</i> , то результат має значення <i>False</i> (в перекл. з англ. Неправда)
=	Перевіряє умову "дорівнює"
>	Перевіряє умову "більше"
<	Перевіряє умову "менше"
<>	Перевіряє умову "не дорівнює"
>=	Перевіряє умову "більше чи дорівнює"
<=	Перевіряє умову "менше чи дорівнює"

(Вираз1) і (Вираз 2) - змінні або константи, що порівнюються даним оператором Частіше доводиться перевіряти не тільки виконання умови, але і їх невиконання Для обробки таких ситуацій використовується оператор *Else*. Його синтаксис виглядає так:

*If* (Вираз1) Оператор (Вираз2) *Then*

(Виконати одні дії)

*Else*

Якщо умова виконується (Істина), то виконуються ті команди, що містяться до оператора *Else*, якщо ні, то ті, що після.

У випадку, коли необхідно перевірити декілька умов, оператор *If ... Then...Else* можна помістити усередині іншого оператора (вкладена перевірка). Вкладені оператори *If... Then* дозволяють виконати серію перевірок у визначеному порядку.

Використання оператора *If... Then...Else* показано у прикладі.

### Структура розгалуження *Select Case... End Select*

Для прийняття складних рішень (особливо при наявності декількох умов) можна користатися блоком *Select Case...End Select*. Його синтаксис простий:

*Select Case X*

*Case Is = (Значення1)*

(Програмні оператори1)

*Case Is = (Значення2)*

(Програмні оператори2)

...

*Case Else*

(Програмні оператори N) (коли не виконана жодна з попередніх умов)

*End Select*

Для кожного оператора *Select Case* у програмі повинен бути відповідний ператор *End Select*. Кількість секцій між *Select Case* і *End Select* може бути вільною.

У операторі *Select Case* значення змінної для якої перевіряється умова та умова повинні мати однаковий тип і можуть бути:

- Оператором порівняння – *Case X < 3*;
- Числом (змінною) - *Case 3*;
- Списком значень - *Case 2, 4, 6*;
- Діапазоном значень – *Case 5 To 10*;
- Строкові рядки

Приклад. Розробити розгалужений алгоритм і програму для обчислення функції

$$Y = \begin{cases} 2.3 \cos(6.6x^2) + 4.7x, & \text{якщо } x \geq 0.3 \\ \sqrt{\frac{3.7 \ln^2 x + 1}{e^{-0.6x} + 1}}, & \text{якщо } x < -1.3 \end{cases}$$

$x = e^{-0.46a \sin(b+a)}$ . Числа *a* і *b* повинні вводитись з клавіатури.

Розміщуємо на формі елементи управління та змінюємо їх властивості так як показано на рис.1.

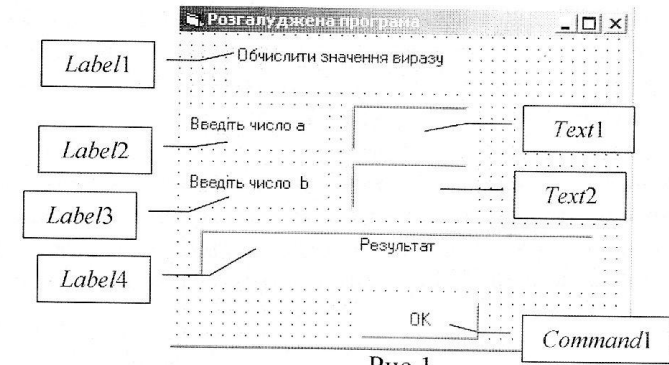


Рис.1

За умови подвійного натиснення мишкою кнопки „OK” переходимо у програми, яке матиме вид (рис.2), результат роботи програми показано на рис.3.

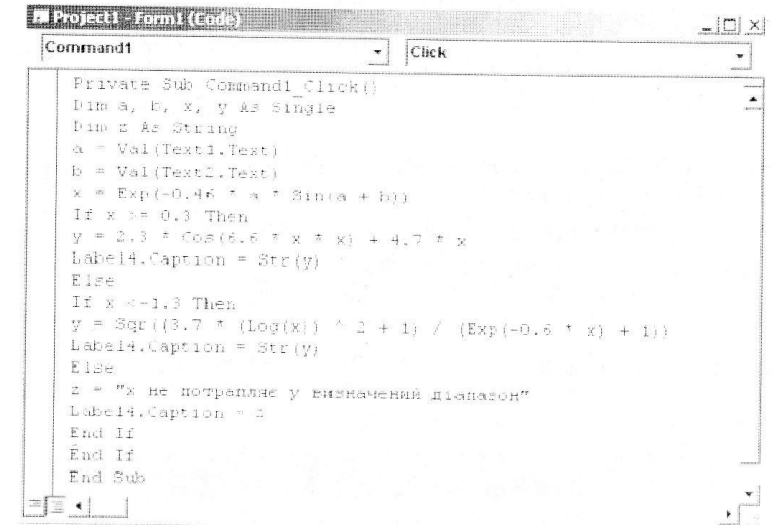
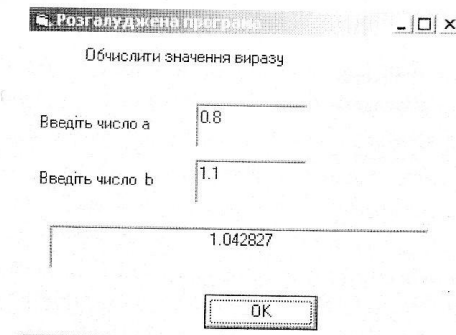


Рис.2



### Налагодження програми. Обробка помилок.

У *Visual Basic* передбачено декілька способів пошуку і корекції помилок під час роботи над проектом. Як правило, помилки, що виникають можна віднести до однієї з трьох категорій.

- Програмний код містить синтаксичні помилки або помилки в загальній структурі використання операторів, що не дозволяє запустити програму на виконання. У цьому випадку *Visual Basic* виводить повідомлення про помилку одразу після її введення. Або повідомлення про помилку з'являється після запуску програми. Такі помилки відносяться до категорії помилок компіляції (*compile errors*), оскільки вони виявляються тоді, коли *Visual Basic* намагається транслювати текст програми в код, що виконується.

- Помилка приводить до того, що програма припиняє роботу на деякій стадії або при спробі виконати деяку операцію. При виникненні такої ситуації *Visual Basic* припиняє виконання програми і виводить повідомлення про помилку. Дуже часто в програмуванні зустрічається помилка – спроба ділення на нуль. Подібні помилки відносяться до категорії помилок часу виконання (*runtime errors*).

- Програма не переривається при виконанні, але результати будуть зовсім не такими (або не зовсім такими), які очікуються. Такі помилки відносяться до категорії логічних помилок (*logical errors*). Виявити й усунути причину цих помилок можна за допомогою налагодження програми.

Налагодженням називається локалізація й аналіз логічних помилок у програмі і відповідна їй корекція.

Для налагоджування програми використовується пункт меню *Debug* або панель інструментів *Debug* (*View - Toolbars - Debug*, рис.4):

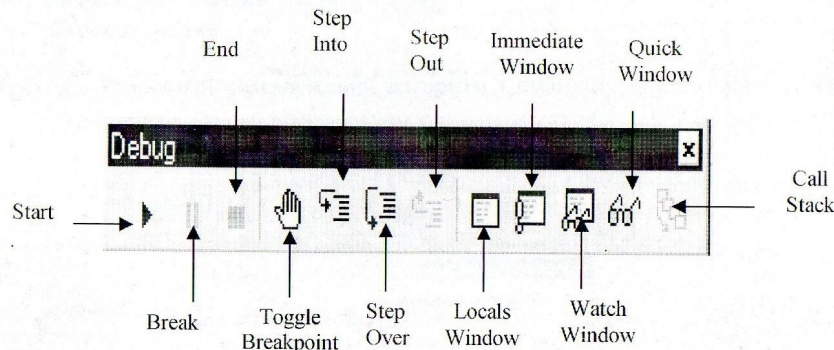


Рис.4

У таблиці 1 подана інформація про піктограми панелі інструментів. Зверніть увагу на те, що і перші піктограми (*Run* (Запуск), *Break* (Пауза) (Кінець)) дублюють аналогічні піктограми стандартної панелі інструментів.

Таб.

### Піктограми панелі інструментів *Debug*

Піктограма	Опис	Клав. викл.
<i>Run</i> (Запуск)	Запускає поточний проект. Після тимчасової зупинки виконує команду <i>Continue</i> (Продовжити)	F5
<i>Break</i> (Зупинка)	Зупиняє виконання поточної програми.	Ctrl+E
<i>End</i> (Кінець)	Завершує виконання поточного проекту.	
<i>Toggle Breakpoint</i> (Точки останова)	Дозволяє позначити рядок тексту програми, на якому необхідно перервати її виконання.	F9
<i>Step Into</i> (Ввійти в процедуру)	Після тимчасової зупинки запускає черговий рядок програмного коду, причому тільки один. Якщо виконуваний оператор є викликом процедури, то наступним буде оператор всередині процедури.	F8
<i>Step Over</i> (Крок через процедуру)	Після тимчасової зупинки запускає черговий рядок програмного коду. Якщо виконуваний оператор є викликом процедури, то за один крок виконується вся процедура.	Shift+
<i>Step Out</i> (Крок до виходу)	Після тимчасової зупинки виконує до кінця поточну процедуру.	Ctrl+Shi
<i>Locals Window</i> (Вікно локальних змінних)	Відкривається вікно <i>Locals</i> . У режимі тимчасової зупинки в ньому виводяться значення змінних, оголошених у поточній процедурі.	
<i>Immediate Window</i> (Вікно безпосереднього виконання)	Відкриває вікно <i>Immediate</i> . У режимі тимчасової зупинки в нього можна ввести деякий вираз на мові <i>Visual Basic</i> і, натиснувши „Enter”, запустити його на виконання.	Ctrl+
<i>Watch Window</i> (Вікно спостереження)	Відкриває вікно <i>Watch</i> . У режимі тимчасової зупинки в ньому виводяться поточні значення вибраних користувачем змінних або виразів	
<i>Quick Watch</i> (Швидке спостереження)	Один з інструментів перегляду в режимі тимчасової зупинки поточного значення вибраної користувачем змінної або виразу	Shift+I
<i>Call Stack</i> (Перегляд стека)	Виводить список послідовності викликів процедур, що привела до виклику поточної процедури	Ctrl+,

У меню *Debug* є ще одна корисна команда – *Run to Cursor*. Вона запускає додаток і зупиняє його в той момент, коли буде досягнута поточна позиція курсору.

### Порядок виконання роботи

1. Відтворити проект, поданий у прикладі 1.
2. Виконати приклад налагодження програми.
3. Скласти розгалужену програму для обчислення значення виразу  $y$  за варіантом.  $x$  та  $a$  можуть набувати довільних значень, і повинні вводитись з клавіатури. В програмі необхідно врахувати випадок, коли значення  $x$  не попадає ні в один числовий відрізок, вказаний у варіанті.

### Варіанти завдань до лабораторної роботи

1	$y = \begin{cases} \frac{\sin(x^2)}{\sqrt{x^2 + a}}, & \text{якщо } x \leq 6 \\ -\cos a \cdot \ln x^2, & \text{якщо } x > 6 \end{cases}$
2	$y = \begin{cases}  x  \cdot \operatorname{tg}(x^2 + a), & \text{якщо } x \leq -6.7 \\ \sqrt{a^2 + \cos(x^4)}, & \text{якщо } x \geq 4.5 \\ x^2 \cdot \ln(2a), & \end{cases}$
3	$y = \begin{cases} 6.7 \cdot \sin \frac{x^7}{9}, & \text{якщо } x \leq 2.3 \\ \ln^2( \sin x  + 0.2), & \text{якщо } x > 3.7 \end{cases}$
4	$y = \begin{cases} \frac{\sin(2x)}{\sqrt{x^2 + 1}}, & \text{якщо } x \leq 3.14 \\ \sqrt{1 + e^{-0.5x}}, & \\ \lg(-0.29x) + \operatorname{tg}(7.8x)^2, & \text{якщо } x > 11.7 \end{cases}$
5	$y = \begin{cases} \sin^5(\lg( a ^a + 5.67)), & \text{якщо } x < 7.89 \\ 5.78a^2 + \frac{\ln^3 2x^a}{1+a^2}, & \text{якщо } x > -7.8 \end{cases}$
6	$y = \begin{cases} \sqrt{\frac{2x^2 + 1}{e^{-0.6x} + 7.2}}, & \text{якщо } x \leq -8.9 \\ \cos^2 x \cdot \operatorname{tg}(0.34x), & \text{якщо } x > 6.7 \end{cases}$
7	$y = \begin{cases} \ln 2( \operatorname{arctg}(2a + 1.3)  + 9.8x), & \text{якщо } x \leq -6.5 \\ \frac{\sqrt{x} \sin a^2}{\sqrt{x + 6.23a^6}}, & \text{якщо } x \geq 5.11 \end{cases}$

8	$y = \begin{cases} \operatorname{tg}(\sin(8.9 +  a + x )), & \text{якщо } x \leq -1.33 \\ \sqrt{\frac{e^{-0.9a} + 8.9x^2}{\operatorname{arctg}x^2 + a}}, & \text{якщо } x > 9.27 \end{cases}$
9	$y = \begin{cases} \frac{a^x \sin 9.8x}{\cos 2x + \ln(a^2 + 87)}, & \text{якщо } x \leq -4.56 \\ 8.3 \cdot \cos x^2 + 3x, & \text{якщо } x > 4.56 \end{cases}$
10	$y = \begin{cases} -6.7 \cdot x^2 + \cos 6.4x, & \text{якщо } x \leq -3.1 \\ \sqrt{1.45x^2 + 7.8e^{ax}}, & \\ \sqrt{1 + \ln^2(a + 2.3)}, & \text{якщо } x \geq 1.37 \end{cases}$
11	$Y = \begin{cases} \sqrt{2.3x^2 +  \cos x }, & \text{якщо } x \leq -1.9, \\ 6.63 - \operatorname{tg}^3 \frac{a}{3}, & \text{якщо } x > 0; \end{cases}$
12	$Y = \begin{cases} \frac{\sin 0.3x}{a^2 + 1}, & \text{якщо } x \leq -2.1, \\ \sqrt{\cos^2 2x + a^2}, & \text{якщо } x > 1.6; \end{cases}$
13	$Y = \begin{cases} \sin 1.3x \cdot \cos^2 \frac{a}{3}, & \text{якщо } x < 1.1, \\ 3.6a + \frac{\ln 3x}{x^3 + 1}, & \text{якщо } x > 4.9; \end{cases}$
14	$Y = \begin{cases} \cos 2a + \sin^3 \frac{x^2}{3}, & \text{якщо } x < 0.8, \\ 1 + \sqrt{\frac{2.3a^2 + 1}{3.1 + e^{-0.9x}}}, & \text{якщо } x > 1.5, \end{cases}$
15	$Y = \begin{cases} \frac{\sin 2x^3}{3\cos a^2 + 2.1}, & \text{якщо } x < 1.4, \\ \sqrt{\frac{3x^2 + 2}{7.7x + \cos 2a}}, & \text{якщо } x \geq 3.3; \end{cases}$

4. Здійснити налагоджування програми. Покрокове виконання і перегляд знач змінних за допомогою пунктів головного меню здійснюється так:

- для закінченого проекту у прикладі 1 виберіть пункти меню *Debug-Watch*, у вікні, що з'явилося введіть значення змінної для спостереження (наприклад, *Label4.Caption*);
- натисніть *F8*, програма запуститься на виконання.
- за допомогою пунктів меню *View-Watch Window* виведіть вік спостережень за змінною у зручне місце на екрані.
- натискайте *F8* та спостерігайте за зміною значень *Label4.Capt*

(результати запишіть у звіті).

- з'ясуйте самостійно як налагоджувати програму за допомогою панелі інструментів *Debug* (запишіть спостереження у звіті).

5. Скласти програму розв'язку економічної задачі за варіантом.

1	<p>Скласти програму підрахунку вартості однієї телефонної розмови для власників стартового пакету „SIM-SIM ювілейний” у мережі UMC в залежності від тривалості (хвилин та секунд) розмови та виду з'єднання. Тарифи у гривнях на одну хвилину подано у таблиці. Тарифікація посекундна з першої секунди. Плата за з'єднання – 0.27 грн. Також врахувати той факт, що згідно із Законом України нараховується збір до Пенсійного фонду – 6% від вартості отриманих послуг зв'язку.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2" style="text-align: center;">Тарифи</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Дзвінки на номери SIM-SIM Сім'ї</td> <td style="text-align: center;">0.10</td> </tr> <tr> <td style="text-align: center;">Дзвінки контрактним абонентам UMC та SIM-SIM</td> <td style="text-align: center;">0.99</td> </tr> <tr> <td style="text-align: center;">Дзвінки в межах України</td> <td style="text-align: center;">2.69</td> </tr> </tbody> </table>	Тарифи		Дзвінки на номери SIM-SIM Сім'ї	0.10	Дзвінки контрактним абонентам UMC та SIM-SIM	0.99	Дзвінки в межах України	2.69
Тарифи									
Дзвінки на номери SIM-SIM Сім'ї	0.10								
Дзвінки контрактним абонентам UMC та SIM-SIM	0.99								
Дзвінки в межах України	2.69								
2	<p>Скласти програму підрахунку суми, що одержить клієнт банку шляхом розміщення ощадних (депозитних) сертифікатів в національній валюті в залежності від суми та терміном збереження, якщо банком встановлені наступні відсотки:</p> <ul style="list-style-type: none"> <li>- термін обігу 90 днів – 10% річних;</li> <li>- термін обігу 180 днів – 12% річних;</li> <li>- термін обігу 360 днів – 15% річних.</li> </ul> <p>Відсотки нараховуються в кінці строку зберігання. Мінімальна сума вкладу 100 грн.</p>								
3	<p>Скласти програму обчислення суми, що одержить клієнт банку, маючи вклад у банку в іноземній валюті в залежності від строку зберігання та суми на рахунку, якщо умови зберігання коштів наступні:</p> <ul style="list-style-type: none"> <li>- строк вкладу 1 місяць – 3.0% річних;</li> <li>- строк вкладу 2 місяці – 4.0% річних;</li> <li>- строк вкладу 3 місяці – 6.0% річних;</li> <li>- строк вкладу 4 місяці – 7.0% річних;</li> <li>- строк вкладу 6 місяців – 8.0% річних.</li> </ul> <p>Відсотки нараховуються в кінці терміну зберігання. Мінімальна сума вкладу \$20.</p>								
4	<p>Скласти програму обчислення відрахувань та прибуткового податку (за ставкою 13%) на заробітну плату, за умов, що у працівника існує лише одна податкова соціальна пільга, така, що якщо сума заробітної плати до відрахувань не перевищує 540 грн., то 61.50 грн. від заробітної плати не обкладається податком. Використати наступні відрахування:</p> <ul style="list-style-type: none"> <li>- якщо заробітна плата менша 150 грн., то у Пенсійний фонд сплачується 1% від неї, якщо – більша 150 грн., то 2%;</li> <li>- 0.5% відрахувань від заробітної плати у Фонд соціального страхування;</li> </ul>								

	<ul style="list-style-type: none"> <li>- 0.5% відрахувань на випадок безробіття;</li> <li>- 1% профспілкових внесків (якщо працівник є членом профсоюзу).</li> </ul>						
5	<p>Скласти програму обчислення коштів (у гривнях), які отримає клієнт банку залежності від суми та терміну зберігання за таких умов:</p> <ol style="list-style-type: none"> <li>1) якщо кошти зберігаються протягом 12 місяців, то процентна стає складає 22% річних;</li> <li>2) якщо клієнт розриває угоду достроково, то одержує кошти то ставці річних за останній місяць.</li> </ol> <p>Відсотки нараховуються щомісячно. Мінімальна ставка не регламентується.</p>						
6	<p>Скласти програму підрахунку вартості однієї телефонної розмови для власників стартового пакету „Джинс” в залежності від тривалості (хвилин та секунд) розмови та виду з'єднання. Тарифи у гривнях на одну хвилину подано у таблиці. Тарифікація посекундна з першої секунди. Плата за з'єднання – 0 грн. Також врахувати той факт, що нараховується збір до Пенсійного фонду 6% від вартості отриманих послуг зв'язку.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tbody> <tr> <td style="text-align: center;">Для абонентів Джинс</td> <td style="text-align: center;">0.50</td> </tr> <tr> <td style="text-align: center;">Мобільні телефони інших операторів</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">Міські телефони України</td> <td style="text-align: center;">2.50</td> </tr> </tbody> </table>	Для абонентів Джинс	0.50	Мобільні телефони інших операторів	1	Міські телефони України	2.50
Для абонентів Джинс	0.50						
Мобільні телефони інших операторів	1						
Міські телефони України	2.50						
7	<p>Складіть програму підрахунку кількості коштів, які одержить клієнт банку залежності від терміну зберігання (в місяцях) та виду вкладу, якщо за вклад гривнях нараховується 18% річних, на вкладах у валюті – 11% річних. Мінімальна сума у гривнях – 100 грн., мінімальна сума в іноземній валюті \$20.</p>						
8	<p>Скласти програму підрахунку вартості друку довільної кількості листів залежності від принтеру та кольору копії, якщо вартість однієї копії лазерному принтері – 0.05 грн., на струйному принтері вартість чорно-білої копії – 0.20 грн., кольорової – 0.40 грн., на матричному – 0.02 грн. Ціна одного листа чистого паперу – 0.05 грн.</p>						
9	<p>Скласти програму підрахунку вартості однієї телефонної розмови в гривнях для власників стартового пакету „ACE” мережі Київстар GSM в залежності тривалості (хвилин та секунд) розмови, часу та виду з'єднання. Тарифи умовних одиниць на одну хвилину подано у таблиці. Тарифікація посекундна Дзвінки тривалістю до 2 секунд включно не тарифікуються. Плата за з'єднання – 0.05 у.о. Також врахувати той факт, що нараховується збір до Пенсійного фонду – 6% від вартості отриманих послуг зв'язку.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tbody> <tr> <td style="text-align: center;">Дзвінки в мережі Київстар GSM</td> <td style="text-align: center;">0.18</td> </tr> <tr> <td style="text-align: center;">Дзвінки по Україні</td> <td style="text-align: center;">0.48</td> </tr> </tbody> </table>	Дзвінки в мережі Київстар GSM	0.18	Дзвінки по Україні	0.48		
Дзвінки в мережі Київстар GSM	0.18						
Дзвінки по Україні	0.48						
10	<p>Скласти програму підрахунку вартості роботи користувача (в грн.) в мережі Internet в залежності від часу та тривалості з'єднання в хвилинах, як провайдером встановлена ціна за роботу з 9<sup>00</sup> до 21<sup>00</sup> – 0.70 у.о. за годину, з 2 до 1<sup>00</sup> та 6<sup>00</sup> до 9<sup>00</sup> – 0.30 у.о., з 1<sup>00</sup> до 6<sup>00</sup> – 0.108 у.о.</p>						
11	<p>Скласти програму для обчислення суми сплати за розрахунково-касовим обслуговування при терміновому переказі грошових коштів у межах Украї</p>						

через банківський телеграф в залежності від суми переказу. Тарифи подано нижче у таблиці.

Тарифи на розрахунково-касове обслуговування в національній валюті

Сума до 200 грн.	3%, але не менше 3.00 грн.
Сума від 200.01 до 600 грн.	2%
Сума від 600.01 до 1000 грн.	1.5%
Понад 1000.01 грн.	1%

12 Скласти програму для обчислення суми сплати за розрахунково-касове обслуговування при терміновому переказі грошових коштів у межах України через банківський телеграф в залежності від суми переказу. Тарифи подано нижче у таблиці.

Тарифи на розрахунково-касове обслуговування в доларах США

Сума до 200 USD	3%, але не менше 5.00 грн.
Сума від 200.01 до 1000 USD	2%
Сума від 1000.01 до 3000 USD	1.5%
Понад 3000.01 USD	1%

13 Скласти програму для обчислення суми сплати за розрахунково-касове обслуговування при терміновому переказі грошових коштів у межах України через банківський телеграф в залежності від суми переказу. Тарифи подано нижче у таблиці.

Тарифи на розрахунково-касове обслуговування в євро

Сума до 200 євро	3%, але не менше 5.00 грн.
Сума від 200.01 до 1000 євро	2%
Сума від 1000.01 до 3000 євро	1.5%
Понад 3000.01 євро	1%

14 Скласти програму підрахунку вартості однієї телефонної розмови в гривнях для власників стартового пакету „BASE” мережі Київстар GSM в залежності від тривалості (хвилин та секунд) розмови, часу та виду з’єднання. Тарифи в умовних одиницях на одну хвилину подано у таблиці. Тарифікація посекундна. Дзвінки тривалістю до 2 секунд включно не тарифікуються. Плата за з’єднання – 0.05 у.о. Також врахувати той факт, що нараховується збір до Пенсійного фонду – 6% від вартості отриманих послуг зв’язку.

Тарифи

Дзвінки в мережі Київстар GSM	
з 8 <sup>00</sup> до 22 <sup>00</sup>	0.24
з 22 <sup>00</sup> до 8 <sup>00</sup>	0.15

15 Скласти програму розрахунку вартості подорожі деяким автомобілем в залежності від кілометражу, ціни та марки палива. Прийняти, що на 100 км пробігу в середньому витрачається 8 л пального. Ціна палива за 1 л:

A98 – 2.45 грн.

A95 – 2.10 грн.

A92 – 1.90 грн.

A80 – 1.65 грн.

Дизельне паливо – 1.70 грн.

Звіт з лабораторної повинен містити:

- тему та мету;

- блок-схеми алгоритмів;

- схематичний вид розташування об’єктів на формі;

- програми;

- висновки.

Контрольні запитання

1. Який синтаксис структури розгалуження *If... Then... Else*?
2. Що таке оператор порівняння? Які слова чи знаки можуть використовувати якості оператора порівняння?
3. Яке призначення має службове слово *Else*?
4. Що таке вкладена перевірка? Наведіть приклади використання вклад перевірок.
5. Як працює блок команд *Select ... End Case*? Чи можна в такій струн використовувати службове слово *Else*?
6. Які категорії помилок є у *Visual Basic*?
7. Як провести налагоджування програми використовуючи головне меню?
8. Як налагоджувати програму за допомогою панелі інструментів *Debug*?



## Лабораторна робота №4. (2 год.)

**Тема:** Використання циклів у *Visual Basic*. Складання програм, що містять цикли.  
**Мета:** Розглянути структури циклів у *Visual Basic*. Навчитись складати циклічні програми.

### Теоретичні відомості

**Цикли** (англ. *Loop*<sup>1</sup>) – це програмні структури, що забезпечують виконання одного чи декількох операторів задану кількість разів. Наприклад, для кожного числа від 1 до 4 необхідно виконати наступні дії: додати 15; помножити на 3 і піднести до 4-го степеня. Для того, щоб не вводити кожний раз число, для якого визначена ця послідовність операцій використовуються структура циклу, яка може бути записана так:

Для кожного  $X$  від 1 до 4 виконати:

$$Y=X+15$$

$$Z=X \times 3$$

$$V=X^4$$

Взяти нове значення  $X$

Процес виконання всіх операторів, що входять в структуру циклу, один раз називається ітерацією (*iteration*). У нашому прикладі здійснюється 4 ітерації. На кожній ітерації виконуватимуться дії:

1) $X=1$ $Y=1+15=16$ $Z=1 \times 3=3$ $V=1^4=1$ $X=2$	2) $X=2$ $Y=2+15=17$ $Z=2 \times 3=6$ $V=2^4=16$ $X=3$	3) $X=3$ $Y=3+15=18$ $Z=3 \times 3=9$ $V=3^4=81$ $X=4$	4) $X=4$ $Y=4+15=19$ $Z=4 \times 3=12$ $V=4^4=256$ $X=5$ (вихід з циклу)
---	--	--	--

Якщо кількість ітерацій визначена, то цикли називаються циклами з фіксованим числом ітерацій (*fixed iteration*). У випадках, коли кількість ітерацій не обмежується (не виконується умова виходу із циклу), цикл називається невизначеним (*indefinite loop*). Блок операторів всередині циклу називається тілом (*body*) циклу.

Найпростішою структурою циклу є фіксовані цикли: *For...Next* і *For Each...Next*. Їх синтаксис має наступний вид:

*For* (лічильник) = (початкове значення) *to* (кінцеве значення) [*Step* (крок)]

(Вираз1)

(Вираз2)

...

(Вираз ...)

*Next* (лічильник)

<sup>1</sup> у лабораторній подано англійський переклад деяких термінів з метою кращого розуміння повідомлень, що формуються під час налагоджування програм

Якщо не задавати значення для кроку, то у наступній ітерації значення лічильника буде змінюватися на 1.

Для прикладу, поданого вище у *Visual Basic* цикл запишеться так:

```
For X=1 to 4
```

```
Y=X+15
```

```
Z=X*3
```

```
V=X^4
```

```
next X
```

Для того, щоб значення лічильника зменшувалося необхідно задати від'ємне значення для кроку.

Цикл *For Each...Next* виконується стільки разів скільки об'єктів є у деякому наборі даних (наприклад, для елементів деякого масиву). Синтаксис циклу має

```
For Each (елемент) In (група)
```

```
(Вираз1)
```

```
(Вираз2)
```

```
...
```

```
(Вираз ...)
```

```
Next (елемент)
```

Також у *Visual Basic* часто використовується цикл *Do...Loop*, який має декілька різновидів:

1) цикл виконується, поки виконується умова. Синтаксис:

```
Do While (умова виконується)
```

```
(Вираз1)
```

```
(Вираз2)
```

```
...
```

```
(Вираз ...)
```

```
Loop
```

Для нашого прикладу структура такого циклу матиме вид:

```
X=1
```

```
Do While X<=4
```

```
Y=X+15
```

```
Z=X*3
```

```
V=X^4
```

```
X=X+1
```

```
Loop
```

2) істинність (правдивість) умови перевіряється вкінці циклу. В такій конструкції оператори циклу виконуються як мінімум один раз. Синтаксис:

```
Do  
(Вираз1)  
(Вираз2)  
...  
(Вираз ...)  
Loop While (умова виконується)
```

Для нашого прикладу цикл матиме вид:

```
X=1  
Do  
Y=X+15  
Z=X*3  
V=X^4  
X=X+1  
Loop While X<4
```

Дві інші структури аналогічні до тих, що ми розглянули за тією відмінністю, що вони перевіряють не правдивість умови, а фальшивість (невиконання умови).  
3) Синтаксис:

```
Do Until (умова не виконується)  
(Вираз1)  
(Вираз2)  
...  
(Вираз ...)  
Loop
```

Для нашого прикладу структура такого циклу матиме вид:

```
X=1  
Do Until X>4  
Y=X+15  
Z=X*3  
V=X^4  
X=X+1  
Loop
```

4) Синтаксис:

```
Do  
(Вираз1)
```

(Вираз2)

...

(Вираз ...)

Loop Until (умова не виконується)

Для нашого прикладу цикл матиме вид:

```
X=1  
Do  
Y=X+15  
Z=X*3  
V=X^4  
X=X+1  
Loop Until X>=4
```

Для виходу із циклу у будь-якому місці використовується команда *Exit*.

Розглянемо використання циклу *For...Next* для вирішення прикладу:

$$y = \sum_{x=1}^2 \frac{\ln(x+1)^{x-1}}{(x+3)^{x+1}}, \text{ step} := 0.1$$

Розмістіть на формі об'єкти та змініть їх властивості так, щоб утворилась форма виду (рис.1):

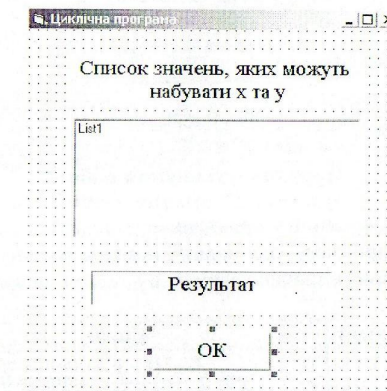


Рис.1

За умови подвійного натиснення мишкою кнопки „OK” переходимо у вікно програми (рис.2):

```

Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
Dim y, x As Single
y = 0
For x = 1 To 2 Step 0.1
y = y + (Log(x + 1) ^ (x - 1)) / ((x + 3) ^ (x + 1))
List1.AddItem ("x = " + Str(x) + "      y = " + Str(y))
Next x
Label12.Caption = y
End Sub

```

Рис.2

Результат роботи програми показано на рис. 3.

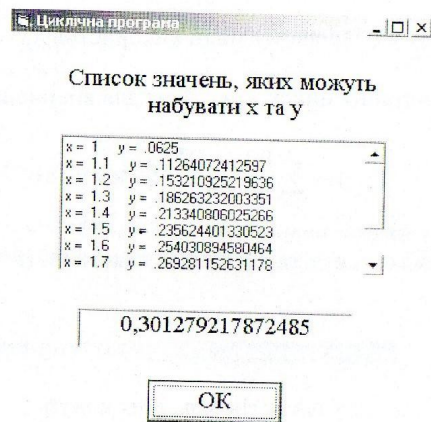


Рис.3

### Порядок виконання роботи

Відтворити проект, поданий у прикладі.

Скласти циклічну програму, для парних номерів варіантів використовуючи цикл *For...Next*, для непарних значень варіантів – цикл *Do...Loop*:

Варіант	Вираз
1	$y = \sum_{x=-2}^2 \frac{x^2(1+x^3)}{10-x^3}, \text{ step} := 0.1$
2	$y = \prod_{x=0.2}^4 \frac{2}{1+e^x}, \text{ step} := 0.2$
3	$y = \sum_{x=0}^{2\pi} \frac{1}{\sqrt[4]{1+\sin^4 x}}, \text{ step} := \pi/10$

4	$y = \prod_{x=0.2}^4 \frac{e^x}{1+x^2}, \text{ step} := 0.1$
5	$y = \sum_{x=0.1}^{1.5} \frac{\text{tg}^4 x}{x}, \text{ step} := 0.2$
6	$y = \prod_{x=0.2}^4 \frac{1}{e^{x \ln x}}, \text{ step} := 0.1$
7	$y = \sum_{x=0}^{2\pi} \frac{\sin^3 x}{\cos^2 x}, \text{ step} := \pi/10$
8	$\prod_{x=0.2}^4 x^2 \ln^3 x, \text{ step} := 0.1$
9	$y = \sum_{x=0.1}^4 \frac{1}{(x^3+1) \cdot \sqrt[5]{(1-x)^2}}, \text{ step} := 0.2$
10	$\prod_{x=1.1}^4 \frac{\sqrt{x}}{\ln^2 x}, \text{ step} := 0.1$
11	$y = \sum_{x=0}^{\pi} \frac{x^x \cdot \sin x^3}{\cos^2 x}, \text{ step} := \pi/2$
12	$\prod_{x=0.2}^4 x^5 \cdot (\sin^2 x - \ln^3 x), \text{ step} := 0.2$
13	$y = \sum_{x=0}^{2\pi} \frac{\sin 3x^4}{\sqrt{1+\cos^4 x}}, \text{ step} := \pi/10$
14	$\prod_{x=1.1}^4 \frac{\sqrt{x^5 - \ln^2 x}}{\ln^3 x}, \text{ step} := 0.3$
15	$y = \sum_{x=0}^{2\pi} \frac{\text{tg}^3 x}{\cos^2 x + \sin^3 x}, \text{ step} := \pi/10$

3. Провести налагоджування програми. Записати у звіт всі значення, яких набувати x та y в процесі виконання програми.

Звіт з лабораторної повинен містити:

- титульний лист;
- мету;
- блок-схему алгоритму;
- схематичний вид розташування об'єктів на формі;
- програму;
- висновки.

### Контрольні запитання

1. Що таке цикли?
2. Що таке ітерація?
3. Які існують види циклів?
4. Наведіть приклади використання циклів для розв'язку економічних задач (задач бухгалтерського обліку, економічного аналізу, тощо).
5. Який синтаксис циклів *For...Next* і *For Each...Next*?
6. Який синтаксис циклів *Do...Loop*?
7. Яка послідовність складання циклічних програм?
8. Як провести налагоджування циклічної програми?

### Лабораторна робота №5. (4 год.)

**Тема:** Обробка масивів.

**Мета:** Навчитись розробляти алгоритми та програми, які виконують дії одновимірними масивами.

#### Теоретичні відомості

Інколи при складанні програм зручно оперувати даними, представлені вигляді таблиць, дані в такому виді називають *масивами*. В такому випадку,  $n$  оперувати групою даних за допомогою одного ім'я і різноманітних індексів, і можливість спростити програму. Є масиви *одновимірні* та *багатовимірні*.

*Одновимірний масив* – це просто список елементів даних. Якщо предст такий масив у вигляді таблиці, то це буде її одна стрічка. Кожний елемент дани зберігається в масиві називається *елементом* масиву. Порядковий номер елемента масиву називається *індексом*. Індекс елемента записується поряд з назвою масиву в круглих дужках без пробілів і може бути лише цілим числом. Найменший порядковий номер масиву називається *нижньою* границею, найбільший – *верхньою*.

Перед використанням масивів їх, так само як і змінні, потрібно описати. Елементи масивів можуть мати такі самі типи, як і змінні. Стрічками

*Dim a(1 To 100) As Integer*

*Dim b(0 To 2) As Single*

*Dim c\_array(1 To 5) As String*

відповідно описуються масив  $a$  зі 100 елементів цілих чисел, масив  $b$  із шістьма елементами дробових чисел, масив  $c\_array$  із шістьма елементами стрічкових даних.

Якщо у програмі потрібно використати якийсь елемент масиву, то до нього потрібно звертатися таким чином: вказується ім'я масиву та індекс, наприклад:

$a(1)$  – дія повинна виконатися з першим елементом масиву  $a$ ;

*For i=1 To 10*

$a(i+1)=a(i)+10$

*Next i*

– у цьому фрагменті кожний наступний елемент масиву  $a$  дорівнює попередньому елементу збільшеному на 10.

Як показано в останньому прикладі для того, щоб одержати доступ до кожного елемента масиву, необхідно використовувати цикли.

Існує ряд алгоритмів для обробки даних масивів. Розглянемо деякі з них.

#### Сортування

Для виконання сортування масив зручно представити у вигляді лінійки елементів в яких зберігаються числа. Наприклад:

**Масив ABC**

-4	16	12	0	-77	11	32	77	-41	2	12	-98	91	значення елемент
1	2	3	4	5	6	7	8	9	10	11	12	13	індекси

При сортуванні, наприклад, за зростанням можна обрати два шляхи:

**Перший метод послідовного пошуку екстремальних елементів:**

Знайти в масиві ABC найменший елемент та його номер і встановити його на 1-ше місце, а 1-й елемент - на місце, де був знайдений найменший елемент. У даному прикладі найменшим є 12-й елемент, а його значення дорівнює 98.

Після перестановки масив буде мати такий вигляд:

**Масив ABC**

-98	16	12	0	-77	11	32	77	-41	2	12	-4	91	значення елементів
1	2	3	4	5	6	7	8	9	10	11	12	13	індекси

Тепер аналогічні дії можна повторити, але пошук мінімального і перестановку треба почати з другого елемента. В даному випадку мінімальним буде п'ятий елемент із значенням -77. Після перестановки масив буде мати такий вигляд :

**Масив ABC**

-98	-77	12	0	16	11	32	77	-41	2	12	-4	91	значення елементів
1	2	3	4	5	6	7	8	9	10	11	12	13	індекси

Таким чином, описані дії треба повторювати доти, доки не буде досягнуто 12-го елемента. Після цього масив ABC буде відсортовано за зростанням.

Отже, якщо розмірність масиву дорівнює N, то треба створити два вкладених цикли, де зовнішній має N-1 разів повторити внутрішній. У внутрішньому циклі треба виконувати пошук найменшого елемента і його індексу. При цьому, після закінчення внутрішнього циклу треба виконувати дві дії:

- робити перестановку знайденого мінімального елемента з елементом, що знаходиться на початку чергового пошуку.

- збільшувати на 1 (одиницю) спеціальну змінну BEG, яка має вказувати на початковий номер елемента, з якого наступного разу буде відбуватись пошук мінімального елемента.

Наприклад:

```

BEG=1 'починаємо шукати з 1 елемента
for i=BEG to 12
  Num_Min=BEG 'припустимо номер мин. елемента = BEG
  Min=ABC(Num_Min) 'а це його значення
  for k=BEG to 12
    if {...} then
      тут відшукаємо номер найменшого елемента в масиві в діапазоні
      'індексів від BEG до 12

```

```

end if
next k
'переставимо значення елемента BEG та Num_Min
Z=ABC(BEG)
'запам'ятаємо значення початкового елемента в Z
ABC(BEG)=ABC(Num_Min)
'переставимо значення MIN в ел. BEG
ABS(Num_Min)=Z 'а значення ел. BEG - в ел. Num_Min
BEG=BEG+1 'зсуємо початок наступного пошуку на 1
next i

```

**Другий метод „бульбашковий“:**

Полягає в тому, що елементи масиву порівнюються *послідовно* та *парно*, і які наступний буде меншим за попередні, то вони міняються місцями. Тобто більший двох елементів „спливає” - просувається в бік зростання значень масиву. Тому цю метод називається „бульбашковим”. Робота починається з першого та друге елементів. У даному випадку перший елемент менший ніж другий, отже, перестанов не виконуємо. Після цього порівнюємо другий і третій елементи. Треба роби перестановку. Тепер масив має такий вигляд:

**Масив ABC**

-4	12	16	0	-77	11	32	77	-41	2	12	-98	91	значення елементів
1	2	3	4	5	6	7	8	9	10	11	12	13	індекси

Для третього і четвертого елементів дії аналогічні.

**Масив ABC**

-4	12	0	16	-77	11	32	77	-41	2	12	-98	91	значення елементів
1	2	3	4	5	6	7	8	9	10	11	12	13	індекси

Ці дії треба повторювати до порівняння 12 та 13 елементів, а потім цю всю процедуру (спочатку) треба повторювати ще 11 разів.

У випадку якщо масив має розмірність n, треба створити 2 вкладених цикли кількістю проходів n-1 кожний. У внутрішньому циклі будуть порівнюватись переставлятися за необхідністю сусідні елементи, а зовнішній має його повторити 1 разів.

Розглянемо приклад сортування масиву за „бульбашковим” методом

1) Розробити алгоритм і програму сортування масиву з 10 елементів, які можуть набувати випадкових цілих значень від 1 до 100.

Розміщуємо на формі об'єкти та відповідно до малюнку змінюємо їх властиво (рис.1)

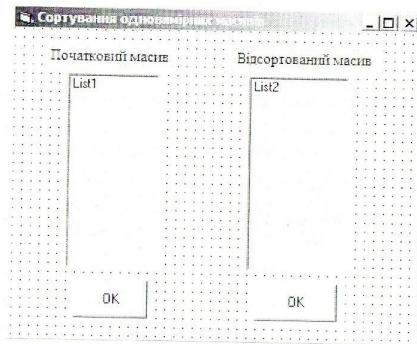


Рис.1

За умови подвійного натиснення мишкою кнопки „OK” переходимо у вікно програми, яке матиме наступний вид (рис.2):

```

Project1 - Form1 (Code)
Command2 Click
Dim x_array(1 To 10) As Integer
Dim i, j, x_p As Integer
Private Sub Command1_Click()
List1.Clear
For i = 1 To 10
Randomize
x_array(i) = Int(100 * Rnd)
Next i
For i = 1 To 10
List1.AddItem x_array(i)
Next i
End Sub
Private Sub Command2_Click()
List2.Clear
For j = 1 To 9
For i = 1 To 9
If x_array(i) > x_array(i + 1) Then
x_p = x_array(i)
x_array(i) = x_array(i + 1)
x_array(i + 1) = x_p
End If
Next i
Next j
For i = 1 To 10
List2.AddItem x_array(i)
Next i
End Sub

```

Рис.2

У процедурі *Command1\_Click()* показано як заповнити у циклі масив, що називається *x\_array* цілими випадковими числами. Функція *Rnd* без аргументу повертає дробове число на проміжку від 0 до 1, помноживши це число на 100 відкинувши дробову частину (функція *Int()*), одержуємо ціле число менше 100. Процедура *Randomize* дозволяє кожний раз після очищення вінка *List1* одержувати нові значення елементів масиву.

У процедурі *Command2\_Click()* здійснюється сортування одержаного масиву.

Результат роботи програми показано на рис.3, слід зауважити, що при кожному новому запуску програми початковий масив буде іншим, тому при перевірці правильності роботи програми слід звернути увагу лише на факт сортування даних, а не на те, які дані відсортовуються.

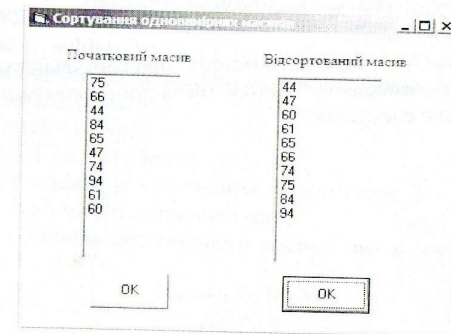


Рис.3

### Порядок виконання роботи

1. Відтворити проект, поданий у прикладі 1.
2. Виконати сортування елементів масиву за зростанням та за зменшенням, використовуючи метод послідовного пошуку екстремальних елементів.

Звіт з лабораторної повинен містити:

- тему та мету;
- блок-схему алгоритмів для обох методів сортування масивів;
- схематичний вид розташування об'єктів на формі до завдання 2;
- програму;
- результат;
- висновки.

### Контрольні запитання

1. Що таке масив даних?
2. Що таке одновимірний масив?
3. Які дані можуть бути записані у масив?
4. Які оператори використовуються для опису масивів?
5. Що таке індекс  $i$  для чого він використовується?
6. Які типи змінних можна використовувати в якості індексів масивів?
7. Яким чином здійснюється доступ до елементів масиву?
8. Чому при роботі з масивами зручно використовувати цикли?
9. Наведіть приклади з життя, в яких можна було б використовувати масиви.
10. Яким чином можна поміняти значення пари змінних у масивах місцями?
11. В чому суть методу послідовного пошуку екстремальних елементів при сортуванні масивів?
12. В чому суть "бульбашкового методу" при сортуванні масивів?
13. Чому в "бульбашковому" методі треба повторювати (скільки разів?) процедуру перестановки елементів?

### Лабораторна робота №6. (2 год.)

**Тема:** Обробка двовимірних масивів.

**Мета:** Навчитись розробляти алгоритми та програми, які виконують дії на двовимірних масивах.

### Теоретичні відомості

Якщо дані можна записати у вигляді таблиці, яка має деяку кількість стовпців та стрічок, то для обробки можна використати двовимірні масиви. Кожний елемент даних, що зберігається в масиві називається *елементом* масиву. Індекс двовимірного масиву складається з двох цифр, перша цифра вказує номер стрічки, друга – номер стовпця на перетині яких міститься потрібний елемент, наприклад запис  $a(5,3)$  означає що нам потрібен елемент, який міститься у 5-ій стрічці та 3-ому стовпці..

Двовимірні масиви описуються подібно до одновимірних. Стрічками

*Dim a(1 To 15, 1 To 10) As Integer*

*Dim b(0 To 2, 0 To 10) As Single*

*Dim c\_array(1 To 5, 1 To 5) As String*

відповідно описуються масиви  $a$  - елементів цілих чисел,  $b$  - елементів дробових чисел, масив  $c\_array$ , який містить стрічкові дані. Виконувати операції з даними двовимірного масиву також зручно використовуючи цикли.

```
For i=1 To 10
  For j=1 To 9
    a(i,j+1)=a(i,j)+10
  Next i
Next j
```

– у цьому фрагменті кожний наступний елемент масиву  $a$  дорівнюватиме попередньому елементу збільшеному на 10.

Розглянемо приклад виконання деяких дій над елементами двовимірного масиву.

Знайти у двовимірному масиві  $a(5, 5)$  усі парні елементи, та замінити їх на одиниці.

Розміщуємо на формі об'єкти та відповідно до малюнку змінюємо їх властивості (рис.1)

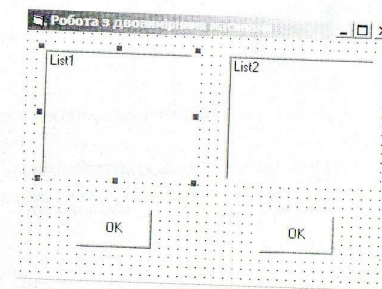


Рис.1

За умови подвійного натиснення мишкою кнопки „OK” переходимо у вікно програми, яке матиме наступний вид (рис.2):

```

Project1 - Form1 (Code)
Command2 Click
Dim a(1 To 5, 1 To 5) As Integer
Dim i, j, z As Integer
Private Sub Command1_Click()
List1.Clear
For i = 1 To 5
For j = 1 To 5
a(i, j) = Int(Rnd * 100)
Next j
Next i
For i = 1 To 5
List1.AddItem (Str(a(i, 1)) + " " + Str(a(i, 2)) + " " + Str(a(i, 3)) + " " +
Next i
End Sub
Private Sub Command2_Click()
List2.Clear
For i = 1 To 5
For j = 1 To 5
z = a(i, j) Mod 2
If z = 0 Then
a(i, j) = 1
End If
Next j
Next i
For i = 1 To 5
List2.AddItem (Str(a(i, 1)) + " " + Str(a(i, 2)) + " " + Str(a(i, 3)) + " " +
Next i
End Sub

```

Рис.2

Результат роботи програми показано на рис.3.

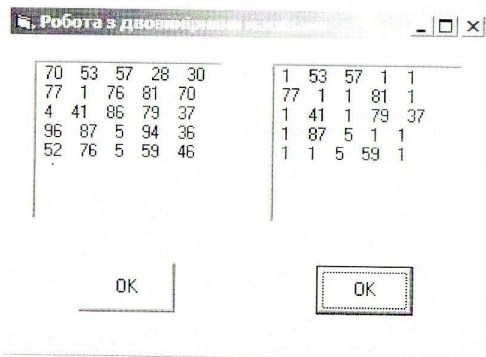


Рис.3

### Порядок виконання роботи

1. Відтворити проект, поданий у прикладі 1.
2. Виконати завдання подане у таблиці за варіантом (варіант вказує викладач).

Вар.	Умова задачі
1	2
1	Знайти середнє арифметичне елементів масиву $A(6,4)$ , відкинувши найбільший та найменший елементи
2	У масиві $A(6,6)$ всі від'ємні елементи, які знаходяться вище головної діагоналі, піднести до квадрату
3	Знайти кількість нульових і суму від'ємних елементів масиву $A(4,5)$
4	В масиві $X(6,6)$ поміняти місцями елементи, які є симетричними відносно головної діагоналі
5	Знайти найменший елемент масиву $X(7,5)$ і записати нулі в ті рядок і колонку, на перетині яких знаходиться найменший елемент
6	Обчислити суму елементів масиву $A(6,6)$ , які знаходяться над головною діагоналлю
7	Обчислити добуток елементів масиву $I(5,5)$ , які знаходяться під головною діагоналлю
8	Переписати елементи головної діагоналі масиву $A(6,6)$ в одновимірний масив $B$
9	У масиві $X(6,5)$ знайти суму елементів 1-го, 3-го і 6-го рядків і розділити на суму елементів 3-ї колонки.
10	В масиві $A(5,6)$ знайти середнє арифметичне елементів для кожного рядка, а потім суму середніх арифметичних
11	Обчислити найбільше і найменше значення в колонках двовимірного масиву $P(5,4)$
12	Обчислити середнє геометричне додатних елементів масиву $X(4,5)$ , які задовольняють умову $X(i,j) > 5$ .
13	Знайти значення трьох найбільших елементів масиву $Y(5,5)$ та їх координати (індекси)
14	У масиві $A(5,6)$ знайти середнє арифметичне найбільшого і найменшого елементів
15	Вивести на екран монітора всі додатні елементи головної діагоналі масиву $A(6,6)$

Звіт з лабораторної повинен містити:

- тему та мету;
- блок-схему алгоритму для власного завдання;
- схематичний вид розташування об'єктів на формі;
- програму;
- висновки.



### Контрольні запитання

1. Що таке двовимірний масив?
2. Які дані можуть бути записані у масив?
3. Які оператори використовуються для опису двовимірних масивів?
4. Як визначити індекс двовимірного масиву?
5. Яким чином здійснюється доступ до елементів двовимірного масиву?
6. Наведіть приклади з життя, в яких можна було б використовувати двовимірні масиви.

### Лабораторна робота №7. (2-4 год.)

**Тема:** Графічні засоби *Visual Basic*. Анімація. Використання графіки для оформлення зовнішнього виду програм.

**Мета:** Ознайомитись з деякими графічними засобами *Visual Basic*.

#### Теоретичні відомості

У *Visual Basic* є 4 елемента управління для роботи з графікою: *Picture Box*, *Image Shape*, *Line*.

Найчастіше для відображення деякої картинки використовується *Picture Box*. Щоб вставити довільну картинку, потрібно для цього елемента змінити властивість *Picture* (обрати файл, що містить потрібний малюнок). Потрібно також знати, що можна розмістити деякий малюнок на фоні форми, для цього потрібно змінити також властивість форми *Picture*. Але і в першому, і в другому випадку картинка, що вставляється матиме свій оригінальний розмір, якщо вона більша, ніж об'єкт, то вона буде обрізана, якщо менша, то займатиме не повний об'єм.

Елемент *Picture Box* має декілька методів, що дозволяють використовувати його для малювання чи друку. А саме, метод *Print* дозволяє виводити текст у вікні *Picture Box*. Для управління характеристиками шрифту використовуються властивості *Picture Box*, які починаються зі слова *Font*.

Для побудови графічних примітивів, найпростіших геометричних фігур – кіл, ліній, точок використовуються відповідно методи: *Circle*, *Line*, *Pset*. Змінюючи властивості *Picture Box* можна змінити вид цих об'єктів (товщину, вид ліній, колір тощо).

Елементи *Image*, *Shape*, *Line* мають лише частину тих властивостей, методів подій, що і *Picture Box*, тому вони потребують менше системних ресурсів завантажуються швидше, ніж *Picture Box*.

Елемент управління *Image* використовується лише для відображення графічних файлів чи деяких об'єктів. На відміну від *Picture Box*, якщо для *Image* встановити *True* властивість *Stretch*, то малюнок, що відображається займе розміри вікна *Image* (буде розтягнутий чи стиснений). Елементи *Shape* та *Line* корисні для малювання геометричних фігур на поверхні форми. Змінюючи властивість *Shape* елемента *Shape* можна обрати вид побудови: прямокутник, квадрат, еліпс, коло, заокруглений прямокутник та заокруглений квадрат, *BorderColor* – колір границь, *FillColor* – колір заповнення області, *BorderStyle* – стиль границі, *BorderWidth* – товщину границі, *DrawMode* – одержати додаткові графічні ефекти. Елемент *Line* подібний до елемента *Shape*, але використовується лише для побудови прямих ліній.

Розглянемо приклад програми, що демонструють графічні можливості *Visual Basic*. Розмістити на формі 11 елементів *Shape* та елемент *Timer*. Змінити властивості для об'єктів *Shape*: *FillColor*, *FillStyle*, *BorderStyle* та *Caption* для *Form1*, елементу *Timer*, властивість інтервал змінити на 1000 (процедура *Timer*, яка відкривається під час подвійного натиснення на значок виконує команди, що містяться в її тілі через інтервал, вказаний у властивостях, інтервал вимірюється в мілісекундах). Одержати форму, показану на рис.1.

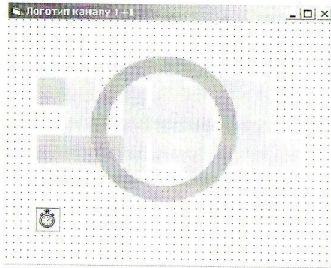


Рис.1

Натиснемо двічі на таймер, щоб перейти до вікна програми (рис.2).

```

Project1 - Form1 (Code)
Timer1
Timer
Private Sub Timer1_Timer()
If Form1.Caption = "Логотип каналу 1+1" Then
Form1.Caption = "Логотип Нового каналу"
Shape1.Visible = False
Shape2.Visible = False
Shape3.Visible = False
Shape4.Visible = False
Shape5.Visible = False
Shape6.Visible = False
Shape7.Visible = False
Shape8.Visible = False
Shape9.Visible = False
Shape10.Visible = False
Shape11.Visible = False
Shape12.Visible = True
Shape13.Visible = True
Shape14.Visible = True
Shape15.Visible = True
Else
Form1.Caption = "Логотип каналу 1+1"
Shape1.Visible = True
Shape2.Visible = True
Shape3.Visible = True
Shape4.Visible = True
Shape5.Visible = True
Shape6.Visible = True
Shape7.Visible = True
Shape8.Visible = True
Shape9.Visible = True
Shape10.Visible = True
Shape11.Visible = True
Shape12.Visible = False
Shape13.Visible = False
Shape14.Visible = False
Shape15.Visible = False
End If
End Sub

```

Рис.2

В результаті роботи програми через кожну секунду логотип одного каналу буде змінюватися на логотип іншого.

### Порядок виконання роботи

1. Відтворити проекти, подані у прикладах.
2. Намалювати будь-який змістовний малюнок із рухомими елементами, показати його викладачу.

*Звіт з лабораторної повинен містити:*

- тему, мету;
- опис виконаної роботи
- висновки.

### Контрольні запитання

1. Які елементи управління використовуються для роботи з графікою?
2. У чому відмінність методів *Picture Box*, *Line* і *Pset*?
3. Як змінити вид ліній та точок (колір, товщину та ін.), що відображаються за допомогою різних методів?
4. Яке призначення елементів управління *Image*, *Shape* і *Line*?
5. Як забезпечити рух (анімацію) елементів у робочій програмі?

### Лабораторна робота №8 (4 год.)

**Тема:** Застосування інструментарію циклів та масивів для складання програм обчислення функцій і побудови графіка.

**Мета:** Навчитись застосовувати структури циклів та оператори роботи з масивами для обчислення функцій. Побудувати графік функції.

#### Теоретичні відомості

Нам потрібно побудувати графік функції для деякого виразу:

$$y = \sqrt{\sin 2x \cdot \left| \operatorname{tg} \frac{x}{2} \right|}, \quad x_{\min} = 0.965, \quad x_{\max} = 3.035, \quad dx = 0.09$$

Для того, щоб відобразити графік цієї функції на екрані потрібно спочатку обчислити значення  $x$  та  $y$ , а потім здійснити такі перетворення, щоб графік займав розміри вікна, в якому він відображається (тобто мінімальна точка по  $y$  на графіку повинна дотикатися до нижньої точки вікна, максимальна по  $y$  - до верхньої точки, крім того, мінімальне значення по  $x$  повинно бути завжди в крайній лівій точці вікна, максимальне значення по  $x$  повинно знаходитись в крайній правій точці вікна), тобто графік потрібно розтягнути або стиснути до розмірів вікна.

Опишемо детальніше як досягнути цієї мети у *Visual Basic*:

1. Потрібно обчислити значення  $x$ -ів та  $y$ -ів і записати їх у відповідні масиви.
2. Знайдемо в масивах значень  $x$ -ів та  $y$ -ів мінімальні та максимальні значення, для того щоб визначити коефіцієнти розтягнення чи стиснення графіку. Якщо різниця між максимальним та мінімальним значенням (по  $x$  чи по  $y$ ) буде менша, ніж ширина чи висота вікна для відображення, то шукаємо коефіцієнт розтягнення графіка, який обчислиться як висота чи ширина вікна поділена на цю різницю, інакше шукаємо коефіцієнт стиснення графіку, і тоді потрібно ділити різницю між максимальним і мінімальним елементом на висоту чи ширину вікна.
3. Оскільки мінімальний елемент по осі  $x$  повинен знаходитись в крайній лівій точці екрану, то цей елемент завжди повинен дорівнювати 0. Це можна зробити якщо відняти від значення мінімального елемента його ж значення, але для того, щоб не втратити співвідношення між іншими елементами масиву потрібно зменшити на цю величину також і інші елементи масиву. При цьому необхідно врахувати те, що  $x$  можуть приймати від'ємні значення.
4. Аналогічним чином розповсюджуємо міркування пункту 3 для значень, що приймає  $y$ .
5. Маючи коефіцієнти розтягнення та стиснення графіку, а також після визначення крайніх точок можна побудувати графік. При побудові графіка слід врахувати той факт, що точка з координатами (0,0) знаходиться в верхньому лівому кутку вікна, а точка з координатами ( $x_{\max}$ ,  $y_{\max}$ ) - у нижньому правому кутку екрану.

Для побудови графіка використаємо *Picture Box*, і метод *Line*.

Послідовність виконання дій буде наступною:

Набрати у редакторі *Microsoft Word (Microsoft Equation)* задану формулу. Розмістити на формі об'єкти (рис.1)

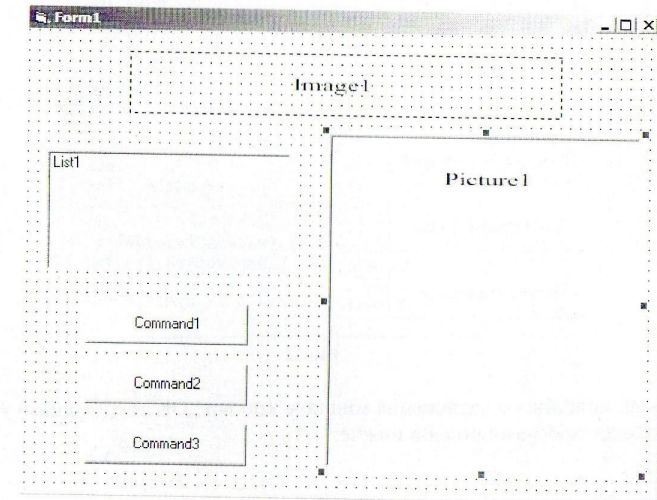


Рис.1

Змінити для об'єктів такі властивості:

Назва	Властивості
Form1	Caption – Табулювання функцій BackColor - ButtonHighlight
Image1	Скопіювати через буфер об'єкт Microsoft Equation Формулу
Command1	Caption – Обчислити значення
Command2	Caption – Побудувати графік
Command3	Caption – Вийти з програми

Отримаємо форму наступного виду (рис.2):

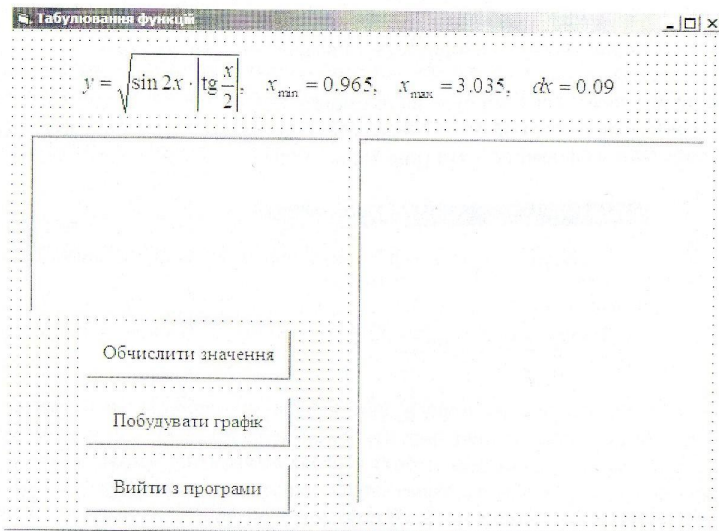


Рис.2

За умови подвійного натиснення мишкою кнопки „OK” переходимо у вікно програми. Текст програми подано нижче:

```
Dim x, y, max_x, min_x, max_y, min_y, koef_x, koef_y As Single
Dim i, el_mas As Integer
Dim S As String
Dim x_mas() As Single
Dim y_mas() As Single
```

```
Private Sub Command1_Click()
List1.Clear
xmin = 0.965
xmax = 3.035
dx = 0.09
i = 0
If ((xmax - xmin) / dx) > Round((xmax - xmin) / dx) Then
    el_mas = Round((xmax - xmin) / dx) + 3
Else
    el_mas = Round((xmax - xmin) / dx)
End If
```

```
ReDim Preserve x_mas(el_mas)
ReDim Preserve y_mas(el_mas)
```

```
For x = xmin To xmax Step dx
    i = i + 1
    y = Sqr((Sin(x)) ^ 2 * Abs(Tan(x / 2))) - 1
    x_mas(i) = x
    y_mas(i) = y
    S = " X = " + Str(x) + "    Y = " + Str(y)
    List1.AddItem (S)
Next x
```

```
End Sub
```

```
Private Sub Command2_Click()
Picture1.Cls
max_x = x_mas(1)
min_x = x_mas(1)
For i = 1 To el_mas
    If max_x < x_mas(i) Then
        max_x = x_mas(i)
    End If
    If min_x > x_mas(i) Then
        min_x = x_mas(i)
    End If
Next i
If min_x < 0 Then
    For i = 1 To el_mas
        x_mas(i) = x_mas(i) + Abs(min_x)
    Next i
Else
    For i = 1 To el_mas
        x_mas(i) = x_mas(i) - Abs(min_x)
    Next i
End If
If (max_x - min_x) < Picture1.ScaleWidth Then
    koef_x = Round(Picture1.ScaleWidth / (max_x - min_x))
Else
    koef_x = Round((max_x - min_x) / Picture1.ScaleWidth)
End If
max_y = y_mas(1)
min_y = y_mas(1)
For i = 1 To el_mas
    If max_y < y_mas(i) Then
        max_y = y_mas(i)
    End If
    If min_y > y_mas(i) Then
        min_y = y_mas(i)
    End If
Next i
If min_y < 0 Then
    For i = 1 To el_mas
        y_mas(i) = y_mas(i) + Abs(min_y)
    Next i
Else
    For i = 1 To el_mas
        y_mas(i) = y_mas(i) - Abs(min_y)
    Next i
End If
If (max_y - min_y) < Picture1.ScaleHeight Then
    koef_y = Round(Picture1.ScaleHeight / (max_y - min_y))
Else
    koef_y = Round((max_y - min_y) / Picture1.ScaleHeight)
End If
x_p = x_mas(1) * koef_x
y_p = Picture1.ScaleHeight - (y_mas(1) * koef_y)
For i = 1 To el_mas
    Picture1.Line (x_p, y_p)-(x_mas(i) * koef_x, Picture1.ScaleHeight - y_mas(i) * koef_y)
Next i
End Sub
```

```
Private Sub Command3_Click()
End
End Sub
```

Результат роботи програми відображається на рис.3

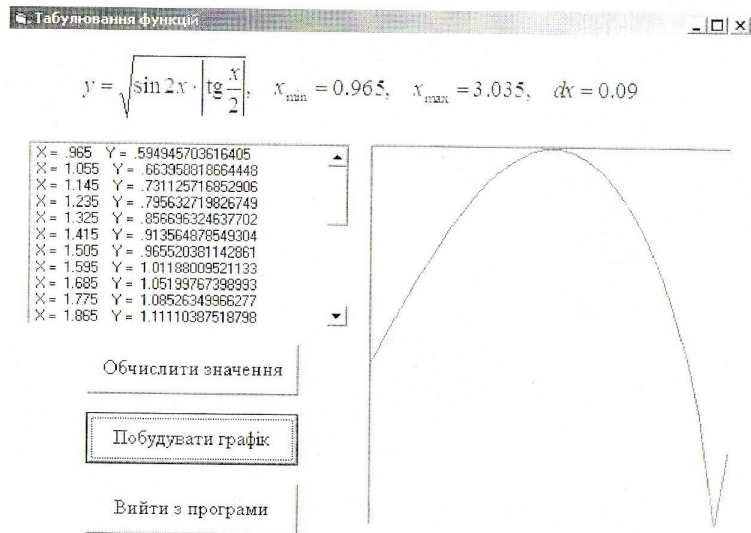


Рис.3

У даному випадку для побудови графіка використовувався метод *.Line*. Для того, щоб відобразити його у вигляді точок потрібен метод *.Pset(x,y), color*, де  $x, y$  – координати точки, що відображається, *color* – номер кольору (число від 0 до 1000000000).

### Порядок виконання роботи

1. Відтворити проект, поданий у прикладі.
2. Скласти програму табулювання функцій (обчислення функції у при зміні  $x$  від  $x_{min}$  до  $x_{max}$  з кроком  $dx$ ) згідно варіанту та побудувати графік спочатку з використанням методу *.Line*, а потім *.Pset*.

Варіант	Вираз
1	$y = \cos^2 \ln( x-1 +2.33)$ , $x_{min} = 0$ , $x_{max} = 2.3$ , $dx = 0.1$
2	$y = 4.36x + \frac{1 + \ln 2x^2}{1 + 2x^2}$ , $x_{min} = 4.7$ , $x_{max} = 7$ , $dx = 0.1$
3	$y = 4.79x^2 + \cos 0.64x^2$ , $x_{min} = -3$ , $x_{max} = 1.6$ , $dx = 0.2$
4	$y = \frac{\sqrt{4.3x^3} \cdot \operatorname{tg} 0.3x}{1 + \ln x^2}$ , $x_{min} = 2$ , $x_{max} = 3.1$ , $dx = 0.05$

5	$y = -0.05x^2 \cdot \ln^2( x +9.4)$ , $x_{min} = -10$ , $x_{max} = 0$ , $dx = 0.5$
6	$y = x^2 \cdot \sin 2x \cdot e^{-0.3x}$ , $x_{min} = -0.3$ , $x_{max} = 0.3$ , $dx = 0.03$
7	$y = \cos x^2 \cdot \frac{\ln(x^2+1)}{\sqrt{x+4.2}}$ , $x_{min} = -2$ , $x_{max} = 2$ , $dx = 0.2$
8	$y = \frac{4.72 \cdot \ln x^2 \cdot \sin x}{2.5 + e^{-0.2x}}$ , $x_{min} = 3$ , $x_{max} = 6$ , $dx = 0.15$
9	$y = 0.34 \cdot \cos \frac{x}{2 + \sin 3x}$ , $x_{min} = 0.1$ , $x_{max} = 0.98$ , $dx = 0.04$
10	$y = \ln^2( \sin x  + x)$ , $x_{min} = 1.5$ , $x_{max} = 3.1$ , $dx = 0.08$
11	$y = e^{-0.23x} + \cos 2.7x$ , $x_{min} = 1.05$ , $x_{max} = 1.25$ , $dx = 0.01$
12	$y = \frac{\sin 2x}{1 + \sqrt{\frac{x^2+1}{1+e^{-0.3x}}}}$ , $x_{min} = 2.07$ , $x_{max} = 2.3$ , $dx = 0.01$
13	$y = \frac{2.3 \cos^2 x}{\sqrt{3.7x^2+x}}$ , $x_{min} = 1.7$ , $x_{max} = 3.3$ , $dx = 0.08$
14	$y = \ln^2( \cos 2x  + 1.3)$ , $x_{min} = 1.5$ , $x_{max} = 2.7$ , $dx = 0.06$
15	$y = \frac{4.7 \cdot \sin 0.2x}{\frac{\sin 2x}{2} + 0.5 \cdot \ln(x^2+1)}$ , $x_{min} = 1.1$ , $x_{max} = 2.02$ , $dx = 0.04$

Звіт з лабораторної повинен містити:

- тему, мету;
- блок-схему алгоритму;
- схему розташування об'єктів на формі;
- програму;
- графік функції;
- висновки.

### Контрольні запитання

1. Яка послідовність складання програми обчислення значень функції та побудови графіка?
2. Як обчислюється коефіцієнт розтягнення чи стиснення графіка?
3. Як помістити будь-яку точку у нижній лівий кут вікна відображення графіку?
4. Яка відмінність методу *.Line* від методу *.Pset*?