

**МІНІСТЕРСТВО АГРАРНОЇ ПОЛІТИКИ УКРАЇНИ
ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ
УНІВЕРСИТЕТ**

**Кафедра вищої математики,
інформатики та ММЕ**

ІНФОРМАТИКА
(Visual Basic)

МЕТОДИЧНІ ВКАЗІВКИ

**для виконання лабораторних робіт
бакалаврами напрямку підготовки 6.030**

Вінниця 2010

Інформатика (*Visual Basic*). Методичні вказівки для виконання лабораторних робіт бакалаврами напряму підготовки 6.030 / Паламарчук Є.А., Яцковська Р.О. - Вінниця: ВНАУ, 2010.- 83с.

Укладачі: Паламарчук Є.А., доцент, к.т.н.
Яцковська Р.О., асистент

Рецензенти: Кравченко Ю.С., к.ф.-м.н., доцент кафедри електроніки ВНТУ
Коляденко С.В. д.е.н., професор кафедри економічної кібернетики ВНАУ

Коротка анотація

Методичні вказівки для виконання лабораторних робіт з дисципліни "Інформатика" містять структурно-модульну схему курсу, розбиття дисципліни на модулі та розподіл балів за категоріями діяльності студента, 8 лабораторних робіт, вимоги до виконання та оформлення лабораторних робіт та список рекомендованої літератури.

Призначені для використання бакалаврами напряму 6.030.

НАУКОВО - МЕТОДИЧНЕ ВИДАННЯ

**Рекомендовано науково-методичною радою
Вінницького національного аграрного університету
протокол № 5 від „22” грудня 2010 р.**

НАУКОВО - МЕТОДИЧНЕ ВИДАННЯ

Паламарчук Євген Анатолійович,
Яцковська Римма Олександрівна,

Інформатика
(*Visual Basic*)

Набір і редагування авторські

Технічний редактор *Олександр Романов*

Верстка

Підписано до друку Формат 60x84/16.
Папір офсетний. Друк різнографічний.
Тираж прим.

Віддруковано у редакційно-видавничому відділі
Вінницького державного аграрного університету
21008, м. Вінниця, вул. Сонячна, 3

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Руденко, В.Д. Практичний курс інформатики: навч.-метод. посіб. / В.Д.Руденко, Макарчук О.М., Патланжоглу М.О. за ред. Мадзігона В.М.АПН України. Ін-т педагогіки. - 2-е вид., перероб. і доп. . - К.: Фенікс, 2001. - 370 с.
2. Інформатика : Комп'ютерна техніка. Комп'ютерні технології. Посіб / За ред. О.І.Пушкаря. - К.: Академія, 2002. - 704 с.
3. Інформатика і комп'ютерна техніка : Навчальний посібник / За ред. М.Є. Рогози. - К.: Академія, 2006. - 368 с.
4. Руденко, В.Д. Практичний курс інформатики: Навч.-метод. посіб. / В.Д.Руденко, Макарчук О.М., Патланжоглу М.О. За ред. Мадзігона В.М.АПН України. Ін-т педагогіки. - К.: Фенікс, 2005. - 304 с.
5. Информатика. Базовий курс : Учебное пособие для студентов высших технических у / С. В Симонович. - 2-е вид.. - СПб.: Питер, 2007. - 639 с.
6. Лігум, Ю.С. Інформатика: Навчальний посібник / Ю.С. Лігум, Г.Ф. Іванченко. - К.: Книжкове вид-во НАУ, 2007. - 111 с.
7. Інформатика для економістів : Навч. посіб. для студ. вузів / В.М. Беспалов, А.Ю. Вакула, А.М. Гострик и др. - К.: ЦУЛ, 2003. - 788 с.
8. Информатика. Комп'ютерна техніка. Комп'ютерні технології : підручник / В.А. Баженов, П.С. Венгерський, В.М. Горлач, І.М. Дудзяний. - 2-е вид.. - К.: Каравела, 2008. - 639 с.
9. Visual Basic 6. Введення в програмування баз даних / Джон Коннэлл - ДМК, Москва , 2000. -720 с
10. Економічна інформатика Практикум: Навчальний посібник/ С.Д. Мамченко, В.А. Одинець. - К.: Знання, 2008. - 710с.
11. Економічна інформатика: Навчальний посібник/ В.С. Григорків, Л.Л. Маханець, Р.Р. Білоскурський и др.; Чернівці: Книги-XXI, 2008. - 464с.
12. Інформатика та комп'ютерна техніка: Навчальний посібник/ М.В. Макарова, Г.В. Карнаухова, С.В. Запара; Ред. М.В. Макарова. - Суми: Університетська книга, 2008. - 665с.

ЗМІСТ

ВСТУП.....	4
Структура орієнтовного розподілу балів за модулями навчальної дисципліни, що присвоюється студентам.....	5
Лабораторна робота № 1	7
Лабораторна робота №2	21
Лабораторна робота №3	31
Лабораторна робота №4	47
Лабораторна робота №5	55
Лабораторна робота №6	63
Лабораторна робота №7	68
Лабораторна робота №8	72

ВСТУП

Однією з головних проблем сьогодення стає широке використання комп'ютерної техніки та комп'ютерних технологій в усіх галузях життєдіяльності людини. У такій ситуації дисципліна «Інформатика» потрібна як основна наука, що формує у людини знання і навички організації спілкування з комп'ютером, роботи з різноманітними, які широко використовуються в бухгалтерському обліку, в оформленні звітних документів різної складності, в проведенні. Дисципліна "Інформатика" призначена для одержання студентами потрібних вмінь складання програм для організації обчислень і т.д.

Метою викладання дисципліни є надання студентам знань теоретичних основ і придбання практичних навичок, розвиток логічного мислення за допомогою мови програмування Visual Basic.

Дисципліна "Інформатика" відноситься до циклу фундаментальних дисциплін освітньо-професійної програми вищої освіти за професійним спрямуванням "Економічна кібернетика", «Облік та аудит», «Фінанси та кредит».

Задачі вивчення дисципліни

У результаті вивчення дисципліни студенти повинні **знати**:

- поняття про програму;
- лінійні програми, структура лінійних програм, принципи виконання програм;
- складання розгалужених програм;
- оператори введення-виведення інформації;
- масиви та їх типи, основи обробки масивів;

На основі отриманих знань студенти повинні **вміти**:

- складати лінійні програми;
- складати розгалужені програми;
- виконувати обробку масивів;
- складати циклічні програми;
- працювати з основними об'єктами ОС Windows;

Звіт з лабораторної повинен містити:

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

Контрольні запитання

1. Яка відмінність методу *.Line* від методу *.Pset*?
2. Що треба зробити, щоб графік функції завжди вміщувався в розміри вікна по горизонталі?
3. Які виникають проблеми при побудові точок функції по вертикалі? Перерахуйте їх і поясніть графіками на папері і власними прикладами.
4. Намалюйте об'єкт *Picture* і покажіть напрямки координат на ньому.
5. Перерахуйте всі міри, які треба застосувати при перетворенні значення функції на її вертикальні координати в вікні.
6. Яка послідовність складання програми обчислення значень функції та побудови графіка?
7. Як обчислюється коефіцієнт розтягнення чи стиснення графіка?
8. Як помістити будь-яку точку у нижню границю вікна відображення графіку?
9. Що треба поміняти в формулі функції $y=f(x)$, щоб вона «перевернулася»?
10. Поясніть на будь-якому графіку, чому при перевертанні функції використовується вертикальний розмір вікна V

$$Y = V - f(x) ?$$

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	$y = \frac{4.7 \cdot \sin 0.2x}{\frac{\sin 2x}{2} + 0.5 \cdot \ln(x^2+1)}$, $x_{\min} = 1.1$, $x_{\max} = 2.02$, $dx = 0.0$

Структура орієнтовного розподілу балів за модулями навчальної дисципліни, що присвоюються студентам

Вид контролю	Модуль	Тема	Навч. заняття (підготовка до виконання)	Виконання індивід. завдань (ОР, реферат, РІР, РР та ін.)	Модульний (змістово-модульний контроль)	Всього балів (сума 4+5+6)
1	2	3	4	5	6	7
Поточний контроль	Введення у предмет. Основи програмування	Лекції: 1. Введення у предмет. Операції з даними. Кодування даних двійковим кодом. Кодування цілих та дійсних чисел. 2. Основні структури даних та змінні. Поняття про програму. Лінійні програми. Структура лінійних програм. Поняття про транслятори. 3. Складання розгалужених програм. 4. Оператори введення-виведення інформації. Поняття про структурування програми. Процедури та функції	1.8			1.8
		Лаб. роботи: 1. Знайомство з навчальною комп'ютерною мережею ВНАУ та ПК студента 2. Управління папками і файлами (Far Manager) 3. Вивчення можливостей оболонки та основних елементів Visual Basic. 4. Типи даних. Математичні оператори та функції VB. Складання лінійних програм. 5. Програми з розгалуженням. Оператори If, Select Case. Налаштування програми. Обробка помилок. 6. Використання циклів у Visual Basic. Складання програм, що містять цикли. 7. Обробка одновимірних масивів.	1.6	2.1	8.7	12.4
		СРС: 1. Методи оперування каталогами (з використанням Провідника і папки Мій комп'ютер). 2. Складання програм з розгалуженими алгоритмами. Складання циклічних алгоритмів. Сортування даних.	1.75	0.1	1.44	3.3
		Всього за модуль 1				17.5

Продовження таблиці

1	2	3	4	5	6	7
Поточний контроль	2 Програмування мовою Visual Basic	Лекції 1. Підпрограми та функції. 2. Локальні та глобальні змінні. Методи організації обміну даними між програмними компонентами. 3. Оператор циклів. Масиви та їх типи. Основи обробки масивів 4. Пошук екстремальних елементів. Сортування 5. Методи керування відтворенням інформації на моніторі. Основи комп'ютерної графіки. Ініціалізація графіки. Графічні оператори. Створення програм з використанням графіки. 6. Електронні компоненти комп'ютерної техніки.	1.8			1.8
		Лаб. роб. 1. Обробка двовимірних масивів 2. Знайомство з графічними засобами <i>Visual Basic</i> . Використання графіки для оформлення зовнішнього виду програм. 3. Застосування інструментарію циклів та масивів для складання програм обчислення функцій і побудови графіка. 4. Колоквіум	1.05	1.4	8.25	10.7
		СРС 1. Використання графіки. Використання циклічних операторів. 2. Дослідження масивів. Дослідження математичних функцій.		0.5	4.5	5
		Всього за модуль 2				17.5
Всього за поточний контроль*						35
Підсумковий контроль (екзамен)						-
Разом						35

Результат роботи програми відображається на рис.3

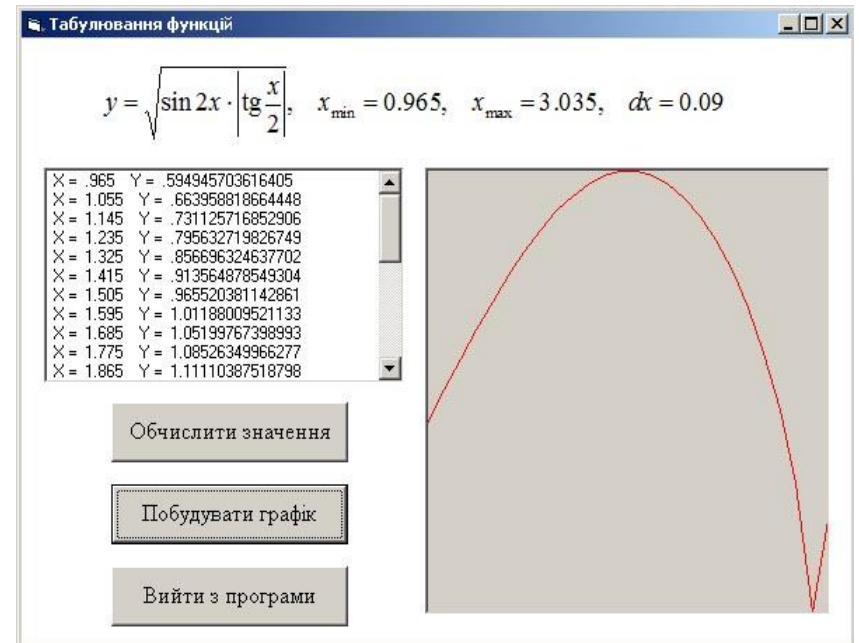


Рис.3

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*.

```

`визначимо у скільки разів треба розтягнути (`або
стиснути) функцію по вертикалі щоби вона мала
розмір висоти вікна об'єкту Picture1
Picture1.ScaleHeight - висота об'єкту в точках

mk = Picture1.ScaleHeight / (ymax - ymin)

`в цьому циклі по точках будуємо графік функції,
яка затабульована в масив y()

For i = 2 To xp

`зробимо так, щоби мінімальна координата по y
`завжди дорівнювала "0"

yy = y(i) - ymin

`а тепер розтягнемо\стиснемо значення функції, щоби
`її максимальне значення завжди дорівнювало висоті
`вікна

yy = yy * mk

`а тут перевернемо функцію. Адже зростання
`координат вікна йде зверху вниз!

yy = Picture1.ScaleHeight - yy

` а тут протягнемо лінію від точки з координатами
`(i - 1, yy) до (i, yy)
`з кольором, що закодований числом 500

Picture1.Line (i - 1, yy)-(i, yy), 500

`переходимо до наступного обрахунку координат по
вертикалі побудови наступної точки ...

Next i

End Sub

Private Sub Command3_Click()
End ' завершуємо програму
End Sub

```

Лабораторна робота № 1

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.

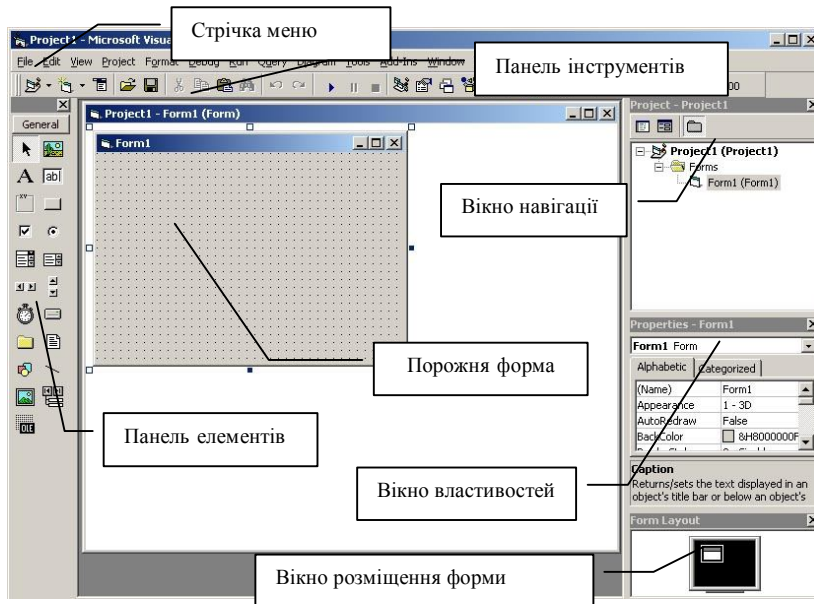


Рис. 2.

```

`тут готуємо стрічку для виведення значень
`аргументу і функції в об'єкт List1
    s = "X = " + Str(x) + "    Y = " + Str(y(i))
    List1.AddItem (s) ' виводимо стрічку
Next i

```

End Sub

```

Private Sub Command2_Click()
`задача цієї підпрограми - побудувати графік
`на підставі значень елементів масиву y()
Dim ymin, ymax, mk As Single

```

```

Picture1.Cls ' чистимо об'єкт Picture1

```

```

`зараз треба знайти мінімальне і максимальне
`значення функції
    ymin = y(1) ' припустимо, що мінімальне
`значення має перший елемент масиву y()
    ymax = y(1) ' аналогічне припущення і щодо
`максимального ...

```

```

` використовуємо один цикл для пошуку
` мінімального і максимального елементів
For i = 2 To xp

```

```

`якщо поточний елемент менше, що ми припускали
`мінімальним

```

```

    If y(i) < ymin Then
        ymin = y(i) ' тоді мінімальним будемо

```

```

`вважати цей

```

```

    End If

```

```

` аналогічно і щодо максимального ...

```

```

    If y(i) > ymax Then
        ymax = y(i)
    End If

```

```

Next i

```


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

```
' Об`являємо масив y, у якого поки що розмірність не визначена.  
' Цей масив, до речі, буде доступним для всіх підпрограм,  
' через те, що ми його об`являємо поза ними.  
  
Dim i, xp As Integer  
Dim y() As Single  
  
Private Sub Command1_Click()  
  
` задача цієї підпрограми - порахувати значення  
` функції у всіх точках `i запам`ятати їх в масив  
y()  
  
Dim x, xmin, xmax As Single  
  
` List1.Clear ` чистимо об`єкт список  
` xmin = 0.965 ` початкове значення аргументу  
функції  
` xmax = 3.3035 ` кінцеве значення аргументу  
функції  
` x = xmin ` починаємо табуляцію з xmin  
` xp = Picture1.Width ` ширина об`єкту Picture1 в  
точках  
  
` а тепер, маючи кількість точок,  
` ми можемо призначити розмірність масиву y()  
ReDim Preserve y(xp)  
  
` dx = (xmax - xmin) / xp ` крок зміни аргументу  
For i = 1 To xp  
` рахуємо значення точки функції для аргументу x і  
заносимо його в y(i)-елемент масиву  
` y(i) = Sqr((Sin(x)) ^ 2 * Abs(Tan(x / 2))) - 1  
` x = x + dx ` одержуємо наступне значення  
аргументу функції
```

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).

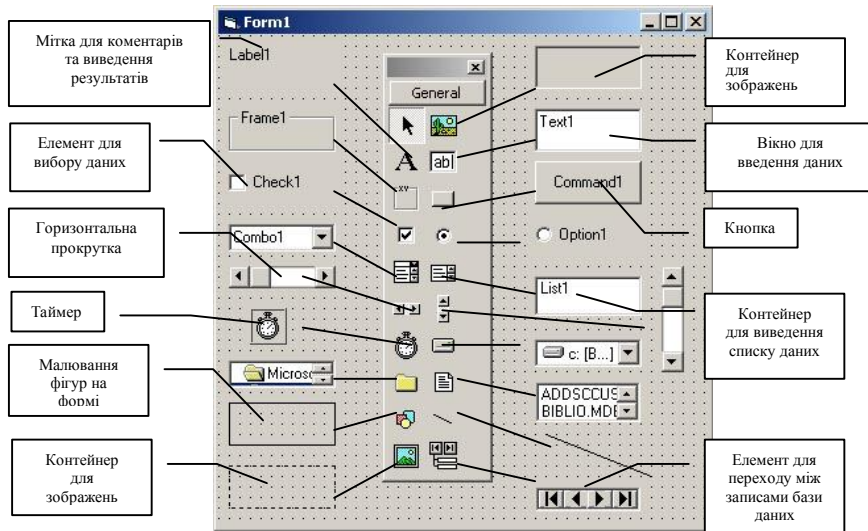


Рис.3.

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, якій виводиться в заголовку форми.

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

ForeColor – кольори тексту.

Height – висота форми.

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

Змінити для об'єктів такі властивості:

Таблиця 1 – Властивості об'єктів

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

Отримаємо форму наступного виду (рис.2):

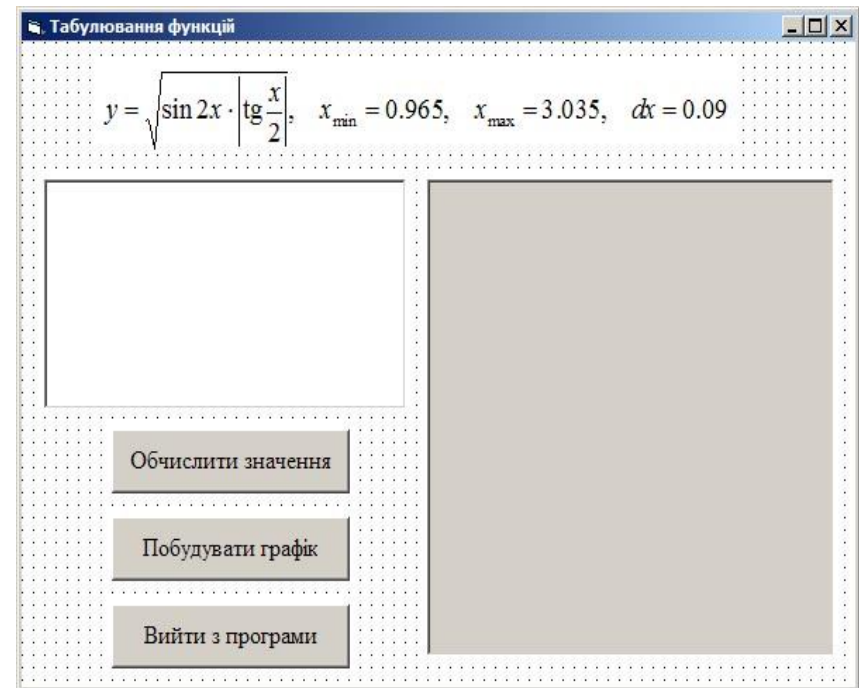


Рис.2

9. І, нарешті, перевертаємо графік. Для цього від максимальної координати V будемо віднімати одержане значення з п.8. Остаточна формула для вертикальної координата така:

$$Y_i = V - (Y(i) - Y_{min}) * K_m$$

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)

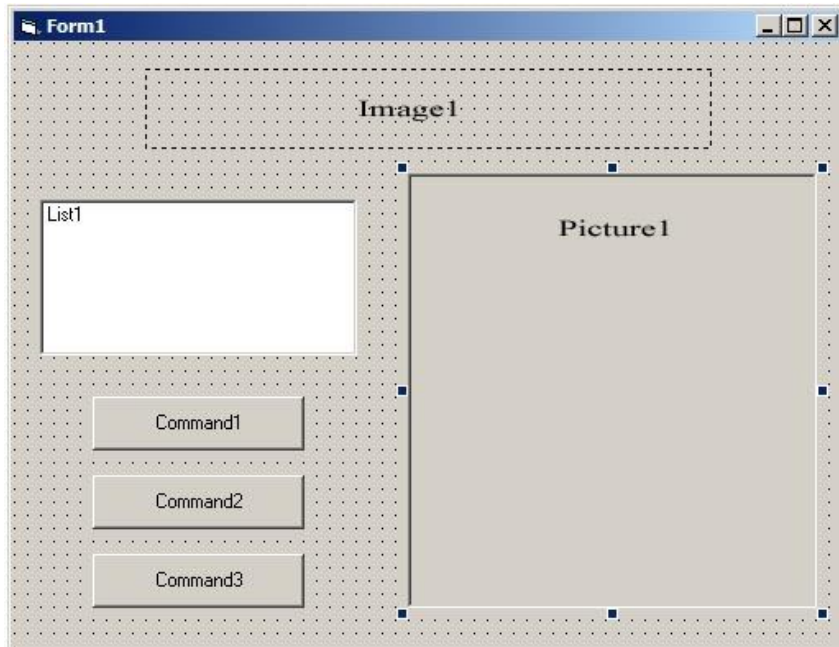


Рис.1

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.

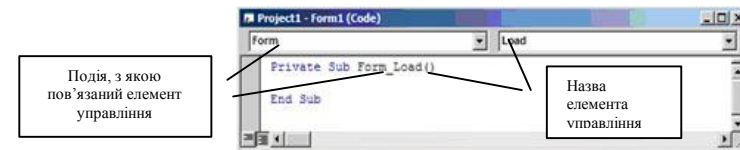


Рис.4.

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
```

} тіло процедури

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 елемента 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).

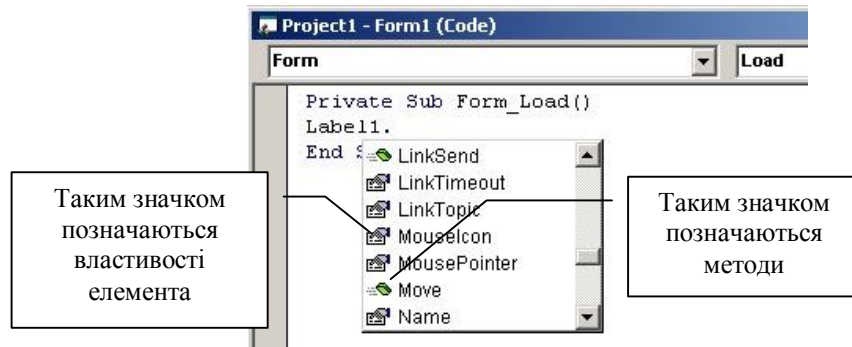



Рис.5.

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);

Опишемо детальніше як досягнути цієї мети:

1. Візьмемо початкове значення аргументу функції, кінцеве його значення, ширину графічного вікна в точках і порахуємо крок зміни аргументу :

$$dx=(X_{max}-X_{min})/N$$

2. Порахуємо значення нашої функції в N точках і кожне значення запам'ятаємо в масив Y(). Таким чином індекси масиву стануть координатами по горизонталі, а значення кожного елемента будуть нести інформацію про координати по вертикалі.

3. Знайдемо в цьому масиві мінімальне і максимальне значення Ymin і Ymax.

4. Порахуємо вертикальний масштабний коефіцієнт, який в подальшому буде «розтягувати» або «стискувати» графік по вертикалі таким чином, щоб його амплітуда завжди дорівнювала розміру вікна V по вертикалі:

$$K_m=(Y_{max}-Y_{min})/V$$

5. Будувати графік будемо лініями, які будуть тягнутись від координат попередньої точки до координат наступної.

6. В процесі побудови будемо перетворювати значення кожного елемента масиву Y(i) на координати цієї точки по вертикалі.

7. Для цього будемо від кожного значення Y(i) віднімати Ymin. Це призведе до того, що графік функції зсунеться (вверх, якщо він має від'ємні точки і вниз, якщо всі точки позитивні) по вертикалі таким чином, що своєю мінімальною точкою торкнеться вісі абсцис, вертикальна координата якої – «0».

8. Кожне таке значення будемо множити на масштабний коефіцієнт, завдяки чому значення максимальної точки графіка стане дорівнювати висоті вікна в точках – V.

Лабораторна робота №8

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:

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

Для того, щоб відобразити графік цієї функції на екрані потрібно спочатку обчислити значення функції в для N-точок, коли аргумент змінюється від X_{\min} до X_{\max} .

Але існує декілька проблем:

1. Треба, щоб графік по горизонталі завжди вміщувався в розміри вікна. Тут прекрасним варіантом є порахувати значення функції в N-точках, де N-кількість точок по горизонталі вікна з графіком.

2. Треба, щоб графік по вертикалі також має займати розміри вікна. Тобто мінімальна його точка повинна торкатись нижньої границі вікна, а максимальна верхньої. При цьому пам'ятаємо, що сама математична функція може приймати взагалі будь-які значення, які можуть бути поза межами розмірів вікна. Наприклад, функція $Y = \sin(x)$ не тільки може приймати від'ємні значення (а на екрані від'ємних координат не існує!), але й амплітуда її складає всього 2 точки. За таких умов на екрані ми практично нічого не побачимо. Отже, маємо якимось чином зсунути функцію по вертикалі, щоб її мінімальне значення дорівнювало «0», а також ще й «розтягнути» її або «стиснути» до вертикальних розмірів вікна.

3. Задача ускладнюється ще й тим, що на екрані вертикальні координати збільшуються не знизу вгору, а зверху донизу! Тобто це означає, що якщо не будуть вжиті спеціальні міри, то графік буде побудованим перевернутим.

- одного вйкового файлу даних для кожної форми з даними про властивості елементів управління на формі (*.frm). 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 (**нагадуємо, 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'явилось 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.

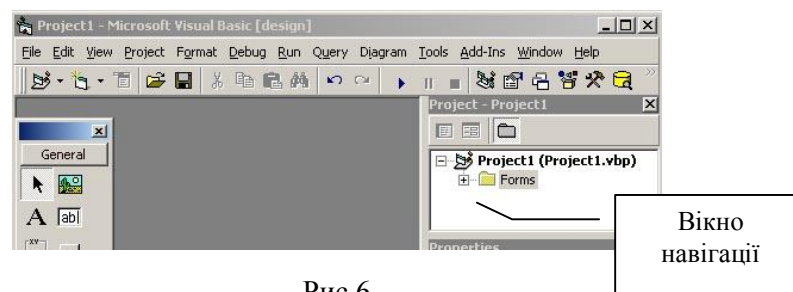


Рис.6

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:

Таблиця 1- Властивості об'єктів

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- <i>Center</i> (цією властивістю визначається положення тексту в комірці: з правого краю, по центру та з лівого краю)
<i>Label1.BorderStyle</i>	1- <i>Fixed Single</i> (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

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

Порядок виконання роботи

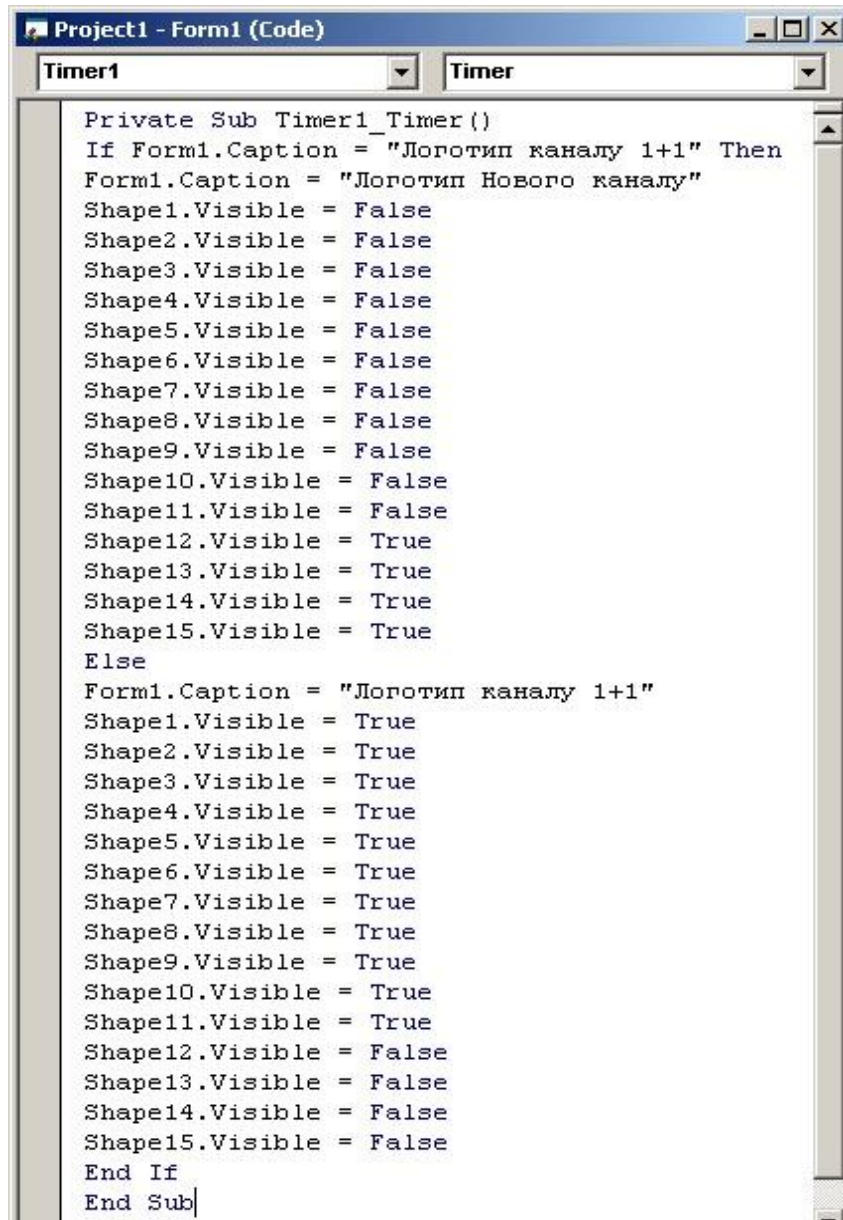
1. Відтворити проекти, подані у прикладах.
2. Намалювати будь-який змістовний малюнок із рухомими елементами, показати його викладачу.

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?



```
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

Форма набуде наступного виду (рис.7)

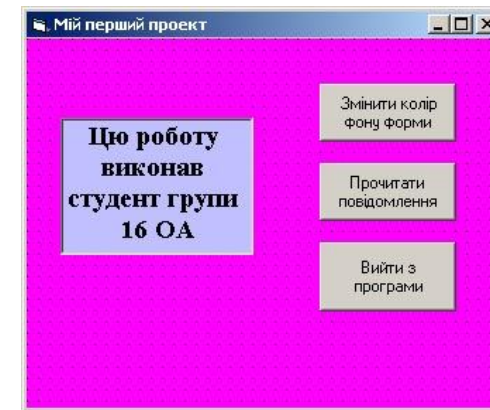
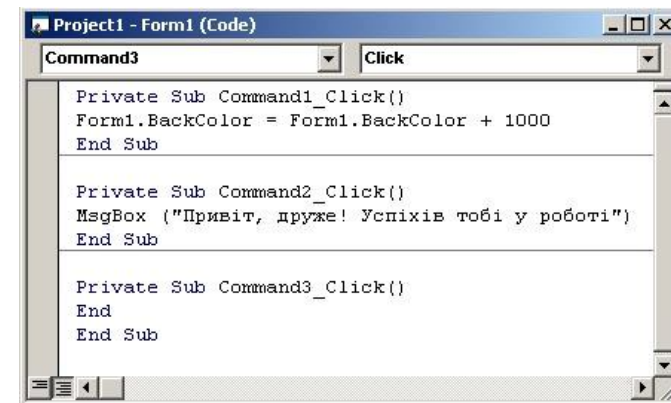


Рис.7.

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”.



```
Project1 - Form1 (Code)
Command3 Click
Private Sub Command1_Click()
Form1.BackColor = Form1.BackColor + 1000
End Sub
Private Sub Command2_Click()
MsgBox ("Привіт, друже! Успіхів тобі у роботі!")
End Sub
Private Sub Command3_Click()
End
End Sub
```

Рис.8.

Запустіть програму на виконання кнопкою  (Start).

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

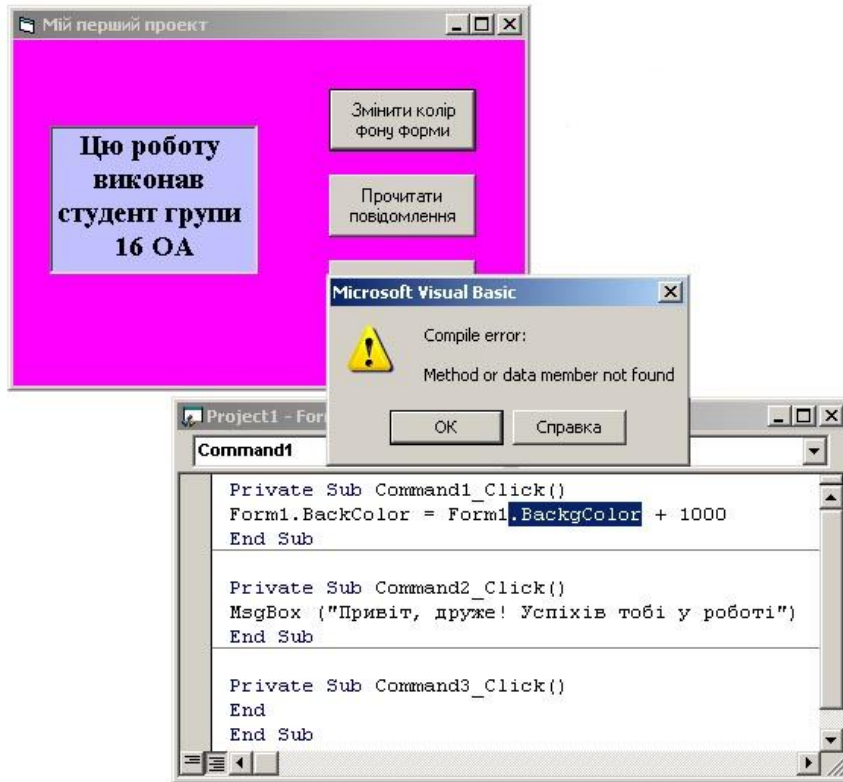


Рис.9.

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'явитись повідомлення типу рис.10.

Stretch, то малюнок, що відображається займе розміри вікна *Image* (буде розтягнутий чи стиснений). Елементи *Shipe* 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 *Shipe*, 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*, елемента *Timer*, властивість інтервал змінити на 1000 (процедура *Timer*, яка відкривається після подвійного натиснення на значок виконує команди, що містяться в її тілі через інтервал, вказаний у властивостях, інтервал вимірюється в мілісекундах). Одержимо форму, показану на рис. 1.

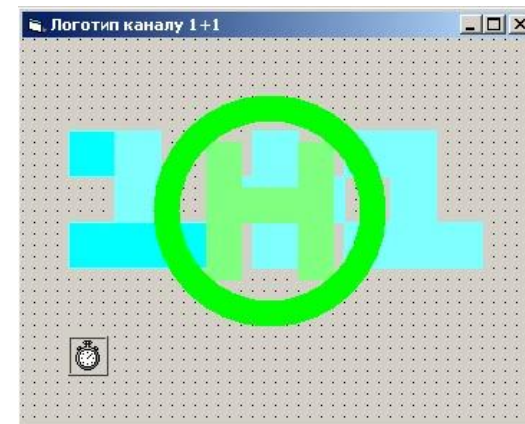


Рис.1

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

Лабораторна робота №7

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 картини 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 і подій, що і *Picture Box*, тому вони потребують менше системних ресурсів і 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 у *True* властивість

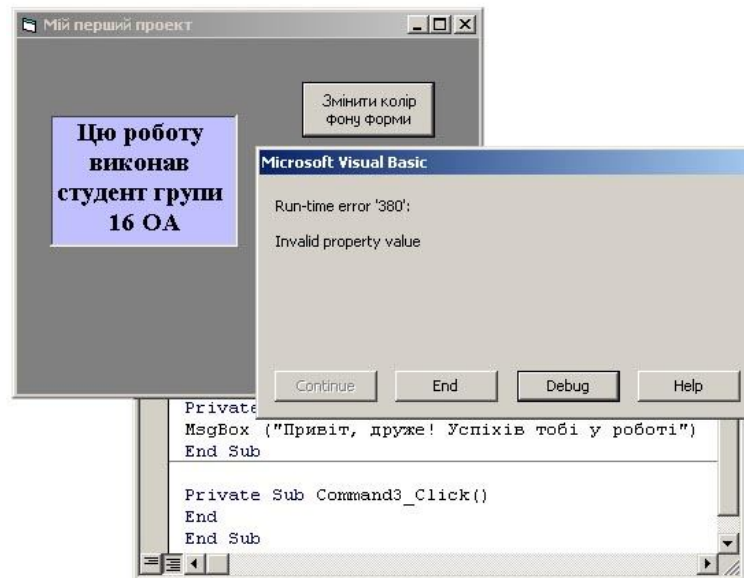


Рис.10

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

Одними із найпоширеніших помилок є:

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

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 елементів *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 2.

Таблиця 2- Властивості об'єктів

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>

	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.

Результат роботи програми показано на рис.2.

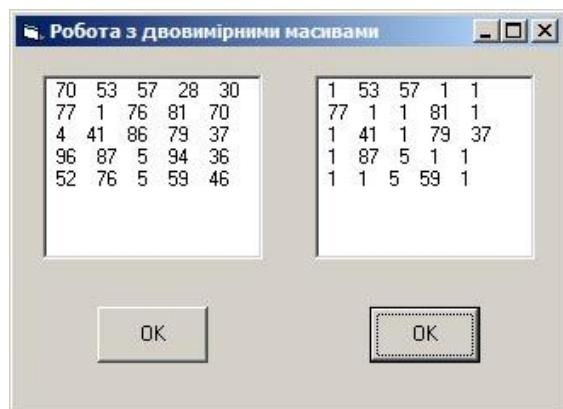


Рис.2

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	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 $H(5,5)$, які знаходяться

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):

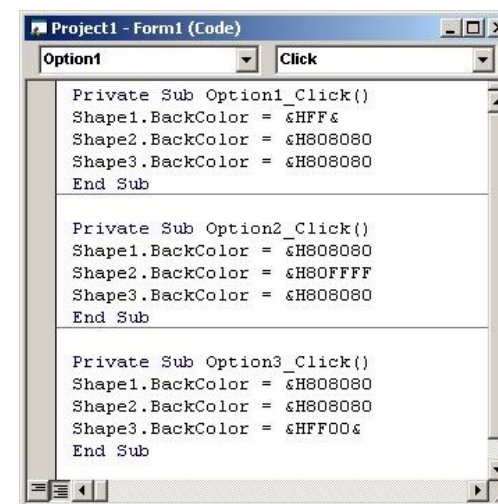


Рис.12

Виконайте програму.

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

Звіт повинен містити тему, мету, короткий опис виконаної роботи та висновки.

Для захисту роботи необхідно продемонструвати викладачу на ЕОМ два 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 елементу?
- 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?

```
        A(i, j) = Int(100 * Rnd) ` одержуємо випадкове  
`число в діапазоні `від 0 до 1, множимо його на 100  
`i беремо цілу частину
```

```
        Next j  
    Next i
```

```
For i = 1 To 10  
    List1.AddItem Str(A(i,1)) + " " + Str(A(i,2)) + "  
" + Str(A(i,3)) + " " + Str(A(i,4)) + " " +  
Str(A(i,5))
```

```
` виводимо значення елементів масиву в список List1
```

```
Next i  
End Sub
```

```
` -----
```

```
Private Sub Command2_Click() ` підпрограма обробки  
`події натискання кнопки Command2
```

```
List1.Clear ` чистимо список
```

```
For j = 1 to 9
```

```
    For i = 1 to 9
```

```
        z = A(i,j) mod 2 ` ділимо елемент масиву  
`націло на 2
```

```
        If z = 0 Then ` якщо залишку немає, то  
`замінімо значення елементу на 1
```

```
            A(i,j) = 1
```

```
        End If
```

```
    Next i
```

```
Next j
```

```
For i = 1 To 5
```

```
    List2.AddItem Str(A(i,1)) + " " + Str(A(i,2)) + "  
" + Str(A(i,3)) + " " + Str(A(i,4)) + " " +  
Str(A(i,5))
```

```
` виводимо значення елементів масиву в список List2
```

```
Next i
```

```
End Sub
```

Лабораторна робота №2

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)

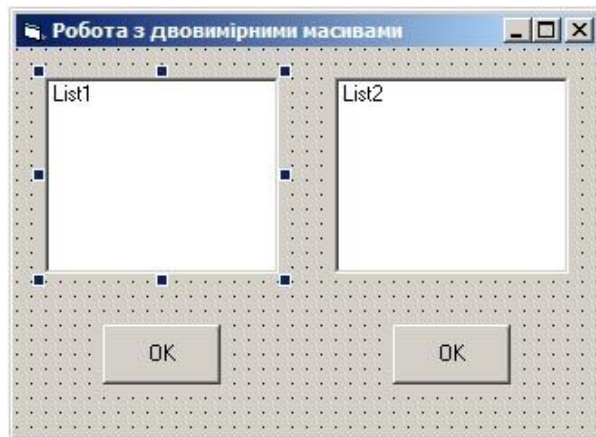


Рис.1

За умови подвійного натиснення мишкою кнопки „OK” переходимо у вікно програми, яке матиме наступний вид:

```
Dim A(1 To 5, 1 To 5) As Single ' об`являємо масив
Dim i, j, z As Integer 'об`являємо змінні цілого
`типу
`-----
Sub Command1_Click() ' підпрограма обробки події
натискання
`кнопки Command1
List1.Clear ' чистимо список
Randomize ' вмикаємо генератор випадкових чисел

For i = 1 To 10
    For j = 1 To 10
```

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 ($x = 'abc', y = 'edf', S = 'abc' + 'edf' = 'abcdef'$) 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 1 подано опис даних цих типів:

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.

Таблиця 1- Числові типи змінних

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

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 $S=x+y$, x і y – цілі числа, значить і результат буде дорівнювати 11, тобто є цілим 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

Лабораторна робота №6

Тема: Обробка двовимірних масивів.

Мета: Навчитись розробляти алгоритми та програми, які виконують дії над двовимірними масивами.

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 j
Next i
```

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

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?

3) припустимо, що $x = "123"$, $y = "абв"$, $S = x + y$, тоді S буде дорівнювати стрічці „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.

Таблиця 2- Функції перетворення

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. Якщо <i>Visual Basic</i> не може перетворити текст на число, то функція повертає 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 у таблиці подано назви та короткий опис основних функцій.

Таблиця 3- Математичні функції

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,

$$y = \frac{\sin^2(\cos^2(x+a))^3}{\ln(cx)} + (tg(x+a)\sqrt{bx})^{2a},$$

якщо відомі x , a , b , c (значення цих змінних вводити з клавіатури).

Розмістіть на формі елементи управління так, як це показано на рис.1 нижче.

У процедурі $Command1_Click()$ показано як заповнити у циклі масив, що називається x_array цілими випадковими числами. Функція Rnd без аргументів повертає дробове число на проміжку від 0 до 1, помноживши це число на 100 і 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.

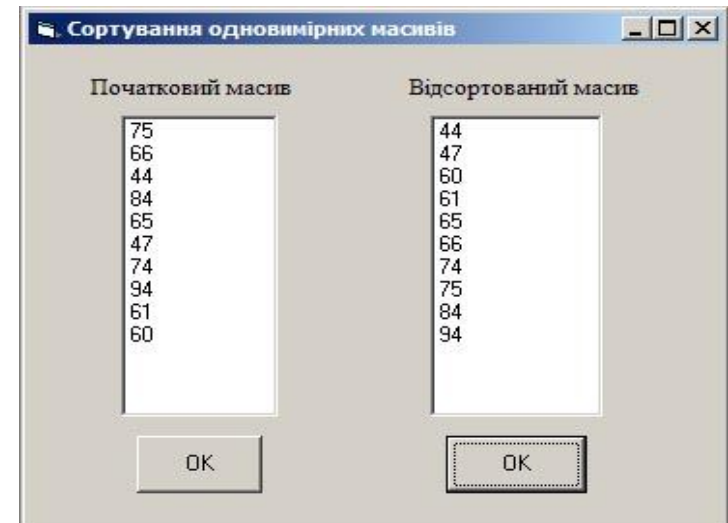


Рис.3

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.


```

Dim X(1 To 10) As Single ' об'являємо масив
Dim i, j, x, p As Integer ' об'являємо змінні цілого
типу
-----
Sub Command1_Click() ' підпрограма обробки події
' натискання кнопки Command1
List1.Clear ' чистимо список
Randomize ' вмикаємо генератор випадкових чисел

For i = 1 To 10
    X(i) = Int(100 * Rnd) ' одержуємо випадкове 'число
в діапазоні 'від 0 до 1, множимо його на 100 'i
беремо цілу частину
Next i

For i = 1 To 10
    List1.AddItem X(i) ' виводимо значення 'елементів
масиву в список List1
Next i
End Sub

-----
Private Sub Command2_Click() ' підпрограма 'обробки
події натискання кнопки Command2
List1.Clear ' чистимо список
For j = 1 to 9
    For i = 1 to 9
        If X(i) > X(i+1) Then ' сусідні елементи
' розташовані не за правилом впорядкування. Тому
' робимо перестановку елементів
            P = X(i)
            X(i) = X(i + 1)
            X(i + 1) = P
        End If
    Next i
Next j

For i = 1 To 10
    List2.AddItem X(i) ' виводимо значення елементів
' масиву в список List2
Next i
End Sub

```



Рис.1

Елементи управління та їх змінені властивості вказані у наступній таблиці 4.

Таблиця 4- Елементи управління та їх властивості

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

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

```

Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
Dim x, a, b, c, y As Single
x = Val(Text1.Text)
a = Val(Text2.Text)
b = Val(Text3.Text)
c = Val(Text4.Text)
y = ((Sin((Cos(x + a) ^ 2) ^ 3)) ^ 2) / Log(c * x) + (Tan(x + a) * Sqr(b * x)) ^ (2 * a)
Label16.Caption = Str(y)
End Sub

```

Рис.2

Рис.3

Порядок виконання роботи

1. Ознайомитись з теоретичним матеріалом.
2. Відтворити поданий у прикладі проект.
3. 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:

У випадку якщо масив має розмірність n , треба створити 2 вкладених цикли з кількістю проходів $n-1$ кожний. У внутрішньому циклі будуть порівнюватись та переставляться за необхідністю сусідні елементи, а зовнішній має його повторити $n-1$ 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 властивості (рис. 1)

Рис.1

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

```

`переставимо значення елемента В та Num_Min
  z = A(b)
`запам`ятаємо значення початкового елемента
`в Z
  A(b) = A(Num)
`перепишемо значення MIN в ел. В
  A(Num) = z `а значення ел.В - в ел. Num_Min
  b = b + 1 `зсунемо початок наступного
`пошуку на 1
Next i

```

Другий метод „бульбашковий”:

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

Масив А

-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 А

-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.

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	$y = \frac{\ln ax \cdot \operatorname{tg} \sqrt{ x+a } + x^{0.3b}}{a \cdot \operatorname{tg} x \cdot e^{\frac{x}{2}} + c \cdot e^{a-x}}$
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.

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 А

-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 А 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 **b**, яка має вказувати на початковий номер елемента, з якого наступного разу буде відбуватись пошук мінімального елемента.

Наприклад:

```

b=1 `починаємо шукати з 1 елемента
for i = b to 12
  Num = b `припустимо, що номер мін.
елемента = b
  Min = A(Num) `а це його значення
  for k = b to 12
    if {.....} then ` тут відшукаємо номер
найменшого `елемента в масиві в діапазоні
індексів від b до 12
    end if
  next k

```

```

For i=1 To 10
    a(i+1)=a(i)+10
Next i

```

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

Як показано в останньому прикладі для того, щоб одержати доступ до кожного елементу 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 А

-4	16	12	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

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

Перший метод послідовного пошуку екстремальних елементів:

Знайти в масиві А 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 А

-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

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 y , але при цьому не ввести значення змінних x, a, b, c ?
10. Якою є послідовність складання лінійних програм?

Лабораторна робота №5

Тема: Обробка масивів.

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

Теоретичні відомості

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

Одновимірний масив – це просто список елементів даних. Якщо представити такий масив у вигляді таблиці, то це буде її одна стрічка. Кожний елемент даних, що зберігається в масиві називається *елементом* масиву. Порядковий номер елемента в масиві називається *індексом*. Індекс елемента записується поряд з назвою масиву у 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 трьох елементів 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 ;

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 можуть 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 №3

Тема: Програми з розгалуженням. Оператори *If*, *Select Case*. Налагодження програми. Обробка помилок.

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

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). **Оператором порівняння** може бути відповідне слово або знак:

Таблиця 1- Оператори порівняння

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

(Виконати інші дії)

End If

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 оператору *If ... Then...Else* показано у прикладі.

Структура розгалуження *Select Case... End Select*

Для прийняття складних рішень (особливо при наявності декількох умов) можна скористатися 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 оператор *End Select*. Кількість секцій між *Select Case* і *End Select* може бути довільною.

Порядок виконання роботи

1. Відтворити проект, поданий у прикладі.
2. 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	$y = \prod_{x=0.2}^4 \frac{e^{\frac{x}{2}}}{1+x^2}, \text{ step} := 0.1$
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	$y = \sum_{x=0}^{\pi} \frac{x^x \cdot \sin x^3}{\cos^2 x}, \text{ step} := \pi/2$

Private Sub Command1_Click()

Dim y, x As Single

y = 0

For x = 1 To 2.1 Step 0.1

y = y + (Log(x + 1) ^ (x - 1)) / ((x + 3) ^ (x + 1))

List1.AddItem ("x = " + Str(x) + " y = " + Str(y))

Next x

End Sub

Рис.2

Результат роботи програми показано на рис. 3.

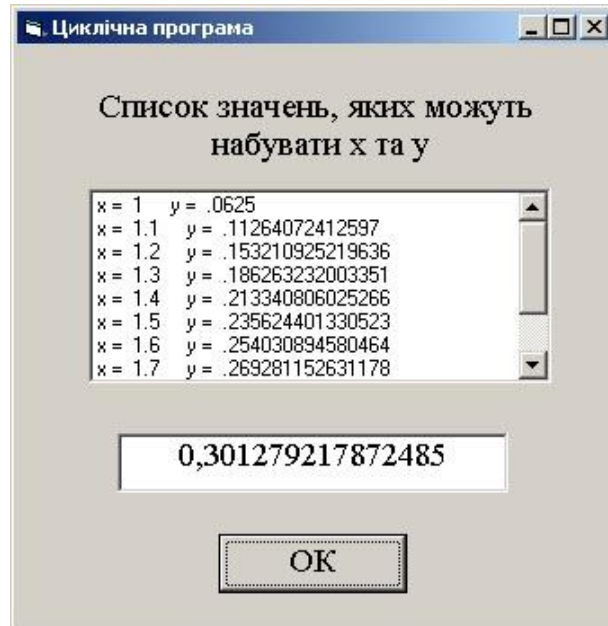


Рис.3

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;
- Стрічкові рядки

Приклад. Розробити розгалужений алгоритм і програму для обчислення функції Y:

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

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

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

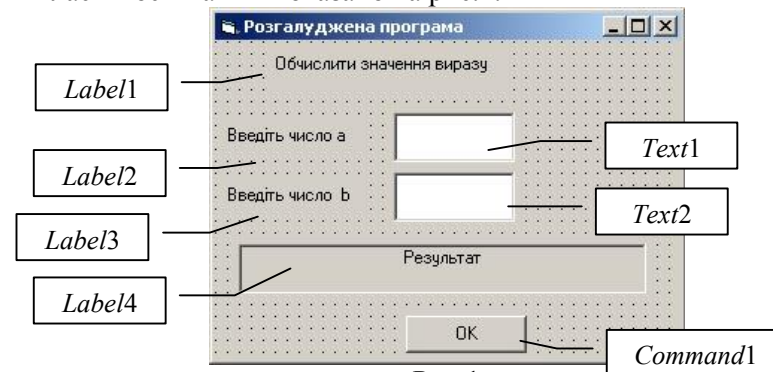


Рис.1

За умови подвійного натиснення мишкою кнопки „OK” переходимо у вікно програми, й розміщуємо програму, що наведено нижче, результат її роботи показано на рис.3.

```

Private Sub Command1_Click()
Dim a,b,x, y As Single ` об'являємо змінні
Dim z As String ` об'являємо змінну
a = Val(Text1.Text) ` Перетворюємо значення поля Text об'єкту
`Text1 на числове
b = Val(Text2.Text) ` Перетворюємо значення поля Text об'єкту
`Text2 на числове

x = Exp(-0.46 * a * Sin(a + b))

If x > 0.3 Then
Y = 2.3 * Cos(6.6 * x * x) + 4.7 * x
Label4.Caption = Str(y) ` Перетворюємо значення у на
`стрічку
Else
If x < -1.3 Then
Y = Sqr((3.7 * (Log(x)) ^2 + 1) / (Exp(-0.6 * x)
+ 1))
Label4.Caption = Str(y) ` Перетворюємо значення у
`на стрічку
Else
z = "x не потрапляє у визначений діапазон"
Label4.Caption = z
End If
End If
End Sub

```

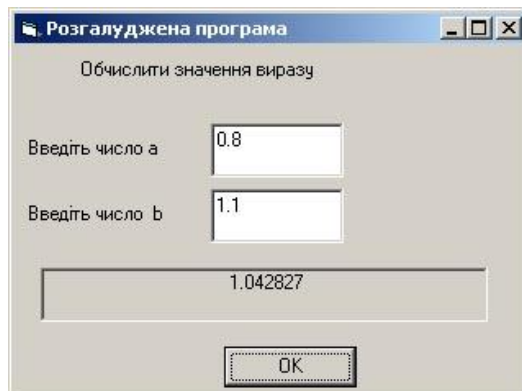


Рис.3

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):

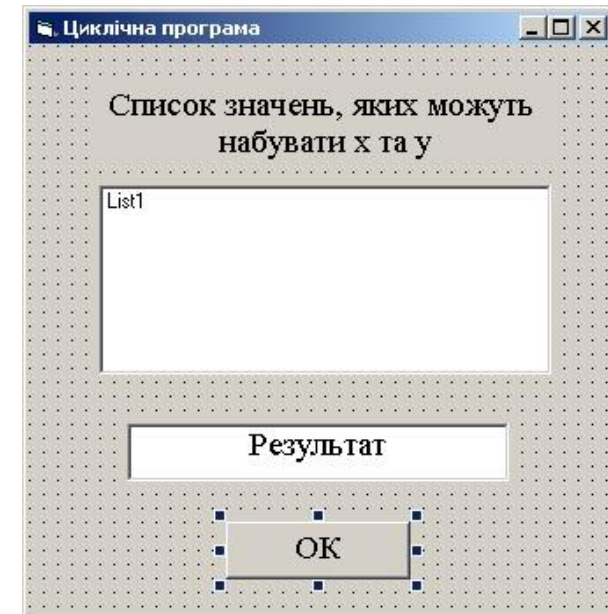


Рис.1

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

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)

...

(Вираз N ...)

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

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):

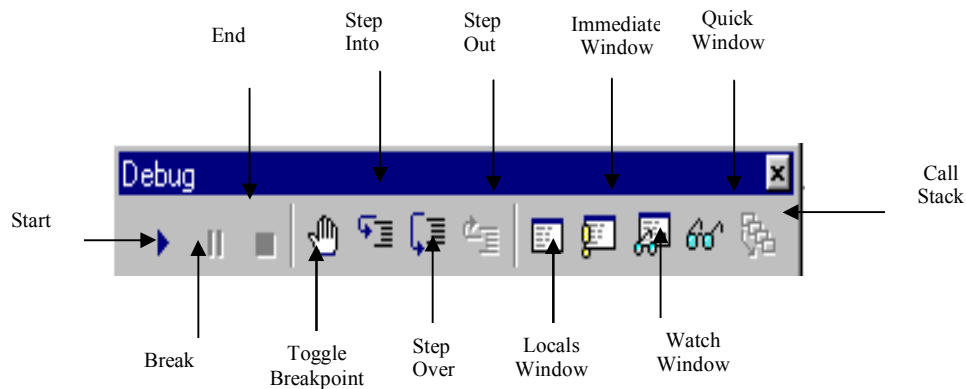


Рис.4

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

Таблиця 2 - Піктограми панелі інструментів *Debug*

Піктограма	Опис	Клавіша виклику
<i>Run</i> (Запуск)	Запускає поточний проект. Після тимчасової зупинки виконує команду <i>Continue</i> (Продовжити)	<i>F5</i>
<i>Break</i> (Зупинка)	Зупиняє виконання поточної програми.	<i>Ctrl+Break</i>
<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.	<i>F9</i>

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 в кінці 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

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)

Якщо не задавати значення для кроку, то у наступній ітерації значення лічильнику 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 вид:

```

For Each (елемент) In (група)
    (Вираз1)
    (Вираз2)
    ...
    (Вираз ...)
```

Next (елемент)

Продовження таблиці 2

<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 проце-дуру)	Після тимчасової зупинки запускає черговий рядок програмного коду. Якщо виконуваний оператор є викликом процедури, то за один крок виконується вся процедура.	Shift+F8
<i>Step Out</i> (Крок до виходу)	Після тимчасової зупинки виконує до кінця поточну процедуру.	Ctrl+Shift+F8
<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 безпосеред-нього 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+G
<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+F9
<i>Call Stack</i> (Перегляд стека)	Виводить список послідовності викликів процедур, що привела до виклику поточної процедури	Ctrl+L

Лабораторна робота №4

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.

Zвіт з лабораторної повинен містити:

- тему та мету;
- блок-схеми алгоритмів;
- схематичний вид розташування об'єктів на формі;
- програми;
- висновки.

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.

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*¹) – 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*).

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

	збір до Пенсійного фонду – 6% від вартості отриманих послуг зв'язку. Тарифи Дзвінки в мережі Київстар GSM 3 8 ⁰⁰ до 22 ⁰⁰ 0.24 3 22 ⁰⁰ до 8 ⁰⁰ 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.

Контрольні запитання

1. Який синтаксис структури розгалуження *If... Then... Else*?
2. Що таке оператор порівняння? Які слова чи знаки можуть використовуватися в якості оператора порівняння?
3. Яке призначення має службове слово *Else*?
4. Що таке вкладена перевірка? Наведіть приклади використання вкладених перевірок.
5. Як працює блок команд *Select ... End Case*? Чи можна в такій структурі 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*?

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	$y = \begin{cases} \frac{ x \cdot \operatorname{tg}(x^2 + a)}{\sqrt{a^2 + \cos(x^4)}}, & \text{якщо } x \leq -6.7 \\ x^2 \cdot \ln(2a), & \text{якщо } x \geq 4.5 \end{cases}$
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	$y = \begin{cases} \frac{\sin(2x)}{\sqrt{x^2 + 1}}, & \text{якщо } x \leq 3.14 \\ \sqrt{1 + e^{-0.5x}}, & \text{якщо } x > 11.7 \\ \lg(-0.29x) + \operatorname{tg}(7.8x)^2, & \text{якщо } x > 11.7 \end{cases}$
5	$y = \begin{cases} \sin^5(\lg(x ^a + 5.67)), & \text{якщо } x < -7.89 \\ 5.78a^2 + \frac{\ln^3 2x^a}{1 + a^2}, & \text{якщо } x > 7.8 \end{cases}$
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	$y = \begin{cases} -6.7 \cdot x^2 + \cos 6.4x, & \text{якщо } x \leq -3.1 \\ \sqrt{\frac{1.45x^2 + 7.8e^{ax}}{1 + \ln^2(a + 2.3)}}, & \text{якщо } x \geq 1.37 \end{cases}$
11	$Y = \begin{cases} \frac{\sqrt{2.3x^2 + \cos x }}{1 + \ln^2 a^2}, & \text{якщо } x \leq -1.9, \\ 6.63 - \operatorname{tg}^3 \frac{a}{3}, & \text{якщо } x > 0; \end{cases}$
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	$Y = \begin{cases} \frac{\sin 2x^3}{3 \cos a^2 + 2.1}, & \text{якщо } x < 1.4, \\ \sqrt{\frac{3x^2 + 2}{7.7x + \cos 2a}}, & \text{якщо } x \geq 3.3; \end{cases}$

12	<p>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.</p> <p>Тарифи на розрахунково-касове обслуговування в доларах США</p> <p>Сума до 200 USD 3%, але не менше 5.00 грн.</p> <p>Сума від 200.01 до 1000 USD 2%</p> <p>Сума від 1000.01 до 3000 USD 1.5%</p> <p>Понад 3000.01 USD 1%</p>
13	<p>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.</p> <p>Тарифи на розрахунково-касове обслуговування в євро</p> <p>Сума до 200 євро 3%, але не менше 5.00 грн.</p> <p>Сума від 200.01 до 1000 євро 2%</p> <p>Сума від 1000.01 до 3000 євро 1.5%</p> <p>Понад 3000.01 євро 1%</p>
14	<p>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</p>

9	<p>Скласти програму підрахунку вартості однієї телефонної розмови в гривнях для власників стартового пакету „ACE” 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.</p> <p style="text-align: center;">Дзвінки в мережі Київстар GSM 0.18</p> <p style="text-align: center;">Дзвінки по Україні 0.48</p>
10	<p>Скласти програму підрахунку вартості роботи користувача (в грн.) в мережі Internet в залежності від часу та тривалості з’єднання в хвилинах, якщо провайдером встановлена ціна за роботу з 9⁰⁰ до 21⁰⁰ – 0.70 у.о. за годину, з 21⁰⁰ до 1⁰⁰ та 6⁰⁰ до 9⁰⁰ – 0.30 у.о., з 1⁰⁰ до 6⁰⁰ – 0.108 у.о.</p>
11	<p>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.</p> <p>Тарифи на розрахунково-касове обслуговування в національній валюті</p> <p style="text-align: center;">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%</p>

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

- для закінченого проекту у прикладі 1 виберіть пункти меню *Debug-Add 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 вікно спостережень за змінною у зручне місце на екрані.
- натискайте F8 та спостерігайте за зміною значень *Label4.Caption* (результати запишіть у звіт).
- з’ясуйте самостійно як налагоджувати програму за допомогою панелі інструментів *Debug* (запишіть спостереження у звіт).

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> <p style="text-align: center;">Тарифи Дзвінки на номери SIM-SIM Сім’ї 0.10</p> <p style="text-align: center;">Дзвінки контрактним абонентам UMC та SIM-SIM 0.99</p> <p style="text-align: center;">Дзвінки в межах України 2.69</p>
---	---

2	<p>Скласти програму підрахунку суми, що одержить клієнт банку шляхом розміщення ощадних (деPOSITНИХ) сертифікатів в національній валюті в залежності від суми та терміном збереження, якщо банком встановлені наступні відсотки:</p> <ul style="list-style-type: none"> - термін обігу 90 днів – 10% річних; - термін обігу 180 днів – 12% річних; - термін обігу 360 днів – 15% річних. <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 рахунок, якщо умови зберігання коштів наступні:</p> <ul style="list-style-type: none"> - строк вкладу 1 місяць – 3.0% річних; - строк вкладу 2 місяці – 4.0% річних; - строк вкладу 3 місяці – 6.0% річних; - строк вкладу 4 місяці – 7.0% річних; - строк вкладу 6 місяців – 8.0% річних. <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"> - якщо заробітна плата менша 150 грн., то у Пенсійний фонд сплачується 1% від неї, якщо – більша 150 грн., то 2%; - 0.5% відрахувань від заробітної плати у Фонд соціального страхування; - 0.5% відрахувань на випадок безробіття; - 1% профспілкових внесків (якщо працівник є членом профсоюзу).

5	<p>Скласти програму обчислення коштів (у гривнях), які отримає клієнт банку в залежності від суми та терміну зберігання за таких умов:</p> <ol style="list-style-type: none"> 1) якщо кошти зберігаються протягом 12 місяців, то процентна ставка складає 22% річних; 2) якщо клієнт розриває угоду достроково, то одержує кошти то ставці 1% річних за останній місяць. <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 – 0.27 грн. Також врахувати той факт, що нараховується збір до Пенсійного фонду – 6% від вартості отриманих послуг зв’язку.</p> <p style="text-align: center;">Для абонентів Джинс 0.50</p> <p style="text-align: center;">Мобільні телефони інших операторів 1</p> <p style="text-align: center;">Міські телефони України 2.50</p>
7	<p>Складіть програму підрахунку кількості коштів, які одержить клієнт банку в залежності від терміну зберігання (в місяцях) та виду вкладу, якщо за вкладу у 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 в залежності від принтеру та кольору копії, якщо вартість однієї копії на 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>