

ФАКУЛЬТЕТ АВТОМАТИКИ, ТЕЛЕМЕХАНІКИ ТА ЗВ'ЯЗКУ

Кафедра «Обчислювальна техніка та системи управління»

МЕТОДИЧНІ ВКАЗІВКИ

**до лабораторних робіт по Visual Basic for Applications
з дисциплін**

***«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»,
«ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА
ПРОГРАМУВАННЯ»»***

Частина 2

Харків 2013

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри «Обчислювальна техніка та системи управління» 31 червня 2011 року, протокол №11.

Рекомендуються для студентів механічного факультету спеціальності 6.050601 «Теплоенергетика» всіх форм навчання.

Укладачі:

доц. І.В. Піскачова,
асист. О.В. Казанко

Рецензент

проф. Г.І. Загарій

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт по Visual Basic for Applications
з дисциплін

*«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ», «ОСНОВИ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ПРОГРАМУВАННЯ»»*

Частина 2

Відповідальний за випуск Єрмакова Л.П.

Редактор Еткало О.О.

Підписано до друку 06.04.12 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 0,75. Тираж 50. Замовлення №

Видавець та виготовлювач Українська державна академія залізничного транспорту,
61050, Харків-50, майдан Фейербаха, 7.

Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

ЗМІСТ

Вступ.....	4
....	4
Лабораторна робота 6. Оператори порівняння у <i>Visual Basic for Applications</i>	5
Лабораторна робота 7. Циклічні обчислювальні процеси у <i>Visual Basic for Applications</i>	12
Лабораторна робота 8. Одновимірні масиви (вектори) даних у <i>Visual Basic for Applications</i>	19
Лабораторна робота 9. Визначення максимального (мінімального) елемента вектора у <i>Visual Basic for Applications</i>	30
Список літератури.....	36

ВСТУП

Мова програмування Visual Basic for Applications (VBA) застосовується для розширення функціональних можливостей Microsoft Office, у якому вона використовується. Наприклад, можна додати власне меню або функцію до вбудованих засобів Excel. Спільно використовуючи VBA з програмами Microsoft Office, можна розв'язувати дуже складні задачі.

Даний курс лабораторних робіт є продовженням "Методичних вказівок до лабораторних робіт по Visual Basic for Applications з дисциплін «Обчислювальна техніка та програмування», «Основи інформаційних технологій та програмування» для студентів механічного факультету. Частина 1" та присвячений вивченню основ програмування в середовищі VBA. Розв'язання різних задач допоможе студентові надалі використовувати отримані навички в професійній діяльності.

ЛАБОРАТОРНА РОБОТА 6

ОПЕРАТОРИ ПОРІВНЯННЯ

У VISUAL BASIC FOR APPLICATIONS

Мета роботи: ознайомитися з формулюванням умов в операторах розгалужених обчислювальних процесів.

Завдання до лабораторної роботи

У даній лабораторній роботі потрібно обчислити усі наведені приклади (6.1-6.3) та індивідуальне завдання. Скласти програму VBA, яка повинна виводити текст умови, що спрацювала, містити кнопку очищення всіх вікон і кнопку виходу із програми.

Короткі теоретичні положення

Visual Basic for Applications для введення порівняльних виразів використовує не тільки загальноприйняті знаки – $>$, $<$, $<>$, $=$, $>=$, $<=$, але й спеціальні логічні оператори.

Логічних операторів шість:

Not (negation) – логічне заперечення;

And (conjunction) – логічне ТА – кон'юнкція;

Or (disjunction) – логічне АБО – диз'юнкція;

Xor (exclusion або exclusive OR) – виключне АБО;

Eqv (equivalence) — еквівалентність;

Imp (implication) — імплікація.

Частіше за інші у програмах використовують And і Or, тому що ці оператори мають універсальне призначення й прості для розуміння. Інші оператори наведеного списку використовуються рідше.

Visual Basic for Applications підтримує комбінації декількох логічних операторів в одному виразі. Порядок виконання операторів регулюється дужками й пріоритетом самих операторів,

інформація про які закладена в мову.

Логічний оператор And

And – логічне ТА - в умовному виразі (кон'юнкція). Цей оператор використовується для запису виразів, які складаються із декількох умов, що повинні виконуватися всі відразу.

Формат параметра And:

```
If Умова1 And Умова2 Then Дія1  
Дія_2
```

Опис формату. Якщо виконується Умова1 і Умова2, то комп'ютер вважає рівним True значення всього умовного виразу й робить Дію1, потім Дію2. Якщо одна або обидві умови не виконуються, то комп'ютер вважає рівним False значення умовного виразу й відразу переходить до Дію2.

Розглянемо задачу про належність числа заданому діапазону.

Приклад 6.1. Ввести X. Якщо його значення належить діапазону від 1 до 10, то сповістити про це користувача.

Розв'язання

Усі вхідні числа повинні задовольняти умову $1 \leq X \leq 10$. Цю умову можна записати інакше – $1 \leq X$ і $X \leq 10$. У програмі даний вираз записується за допомогою оператора And:

```
X = Val(TextBox("Введіть X"))  
If 1 <= X And X <= 10 Then  
    txtresult.Text = "Належить інтервалу"  
Else  
    txtres.Text = "Не належить інтервалу"  
End If
```

Робота програми. Аналіз умовного виразу дає True тільки в тому випадку, якщо перша і друга умови виконуються. У всіх інших випадках результатом буде False (виведення на екран повідомлення - не належить інтервалу).

Робота умовного оператора And наведена в таблиці

істинності. У ній розглядаються тільки дві умови, розділені And, але їх може бути й більше, що не впливає на логіку роботи оператора.

Значення True може бути отримане тільки в одному випадку – при виконанні всіх умов виразу (таблиця 6.1). Це головна особливість оператора And. Саме тому And використовують у тих алгоритмах, у яких відслідковується одна ситуація — виконання всіх умов.

Таблиця 6.1 – Таблиця істинності оператора And

Умова1	Умова2	Умова1 ТА Умова2
True	True	True
True	False	False
False	True	False
False	False	False

Операторів And, як і всіх інших логічних операторів, в одному виразі може бути декілька. Їх кількість обмежується лише розміром рядка.

Логічний оператор Or

Or – логічне -АБО (диз'юнкція) в умовному виразі використовується для запису декількох умов, будь-яка кількість із яких повинна виконатися. Під словом будь-яка кількість мається на увазі виконання однієї, декількох або всіх умов.

Робота оператора Or для двох умов наведена в таблиці істинності (таблиця 6.2).

Таблиця 6.2 - Таблиця істинності оператора Or

Умова1	Умова2	Умова1 АБО Умова2
True	True	True
True	False	True
False	True	True
False	False	False

Отже, значення False виходить тільки в одній ситуації, коли всі умови не виконуються. Якщо вираз складається із двох умов, то

значення True буде відповідати виконанню однієї або обох умов.

З таблиці 6.2 ми бачимо, що значення True у виразах з оператором Or виходить у будь-якій ситуації, крім випадку, коли всі умови не виконуються.

Формат параметра Or:

```
If Умова1 Or Умова2 Then Дія1  
Дія2
```

Опис формату. Якщо одна АБО дві умови виконуються, то комп'ютер робить Дію1, потім Дію2. Якщо жодна умова не виконується, комп'ютер відразу переходить до Дію2.

Приклад 6.2. Увести значення x та якщо x дорівнює 3 або більше 8 обчислити вираз:

$$y = \frac{x^2}{1+x^6}.$$

Фрагмент програми:

```
If x = 3 Or x > 8 Then y=x^2/(1+x^6)  
txtres.Text = y
```

Робота програми. Обчислення у виконується у тому випадку, якщо змінна x буде дорівнювати 3 або більше 8.

Приклад 6.3. Для X , що НЕ належить інтервалу від 4 до 9, обчислити вираз $S=X^3$.

Фрагмент програми:

```
If X < 4 Or X > 9 Then S = X^3  
txtres.Text = S
```

Робота програми. Вираз буде обчислюватися тільки при змінній X , яка менше 4 або більше 9. Якщо змінна входить в інтервал від 4 до 9, то вираз обчислюватися не буде.

Порядок виконання лабораторної роботи

- 1 Запустити програму Visual Basic for Applications.
- 2 Вивести на форми кнопки, надписи й текстові вікна, як того потребують приклади (6.1-6.3).
- 3 Присвоїти елементам відповідні властивості.
- 4 Провести розрахунки.

Зміст звіту


У звіті повинно бути:

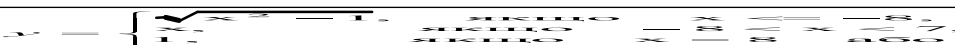
- 1 Визначена мета завдання.
- 2 Наведено рисунки з елементами керування: форма із кнопками, текстове вікно, надписи.
- 3 Перелічені основні властивості використаних об'єктів керування.
- 4 Висновки по лабораторній роботі.



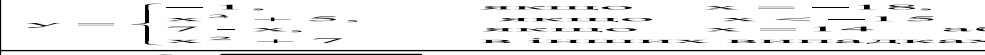
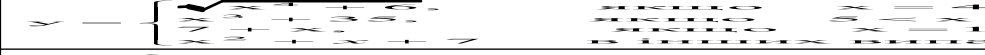


Питання до захисту лабораторної роботи 6.


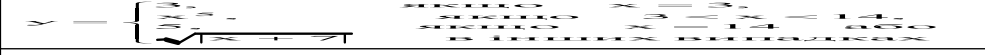

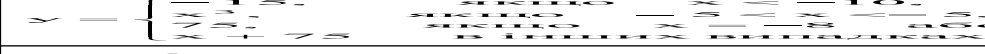
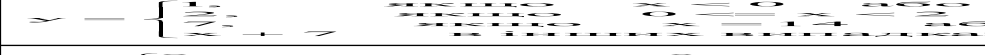
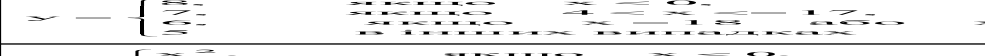
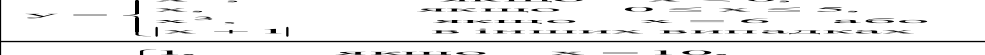
- 1 Які оператори використовуються в Visual Basic for Applications для введення складних умовних виразів?
- 2 Навести формат оператора And.
- 3 Навести таблицю істинності оператора And.
- 4 Навести приклад використання оператора And.
- 5 Навести формат оператора Or.
- 6 Навести таблицю істинності оператора Or.

Варіанти індивідуальних завдань до лабораторної роботи 6

Варіант	Завдання
1	

Варіант	Завдання
2	

3	
4	
5	
6	
7	
8	

Варіант	Завдання
9	
10	
11	
12	
13	
14	
15	

ЛАБОРАТОРНА РОБОТА 7

ЦИКЛІЧНІ ОБЧИСЛЮВАЛЬНІ ПРОЦЕСИ У VISUAL BASIC FOR APPLICATIONS

Мета роботи: ознайомитися з оператором циклу For...Next.

Завдання до лабораторної роботи

У даній лабораторній роботі потрібно обчислити усі наведені приклади (7.1 – 7.3) та потрібно скласти програму розрахунків математичних прикладів (варіанти завдань вказує викладач), що містить суму й добуток числового ряду.

Короткі теоретичні положення

Оператори циклу застосовуються для розв'язання задач, у яких потрібно багаторазово виконувати деякі дії (оператори). В

VBA використовуються три оператори циклу: For...Next, Do...Loop і While...Wend. Перший зручний у тих випадках, коли заздалегідь відомо скільки разів потрібно виконати дію. Другий і третій застосовують у тих алгоритмах, коли не відомо скільки разів повинен відробити цикл і програма виконує дії, поки не буде виконана певна умова.

Оператор For...Next (параметричний цикл).

Його формат виглядає так:

```
For лічильник = Поч_значення To Кінц_значення [Step знач_кроку]
    [Блок операторів]
[Exit For]
Next [лічильник]
```

For...Next повторює блок операторів доти, поки значення змінної - лічильника не стане більше (а якщо початкове значення більше кінцевого, то менше) кінцевого значення циклу. Після кожної ітерації (чергового виконання блока операторів) до значення змінної - лічильника додається значення кроку. Якщо крок дорівнює 1, то рядок Step 1 можна не писати.

Лічильник – числова змінна, значення якої автоматично змінюється в процесі роботи й відслідковується циклом.

Поч_значення – початкове значення лічильника циклу. Воно може бути задане числом, змінною, математичним виразом. Значення може бути цілим або дробовим, додатним або від'ємним.

Кінц_значення – кінцеве значення лічильника. Воно може бути задане числом, змінною, математичним виразом. Значення може бути цілим або дробовим, додатним або від'ємним.

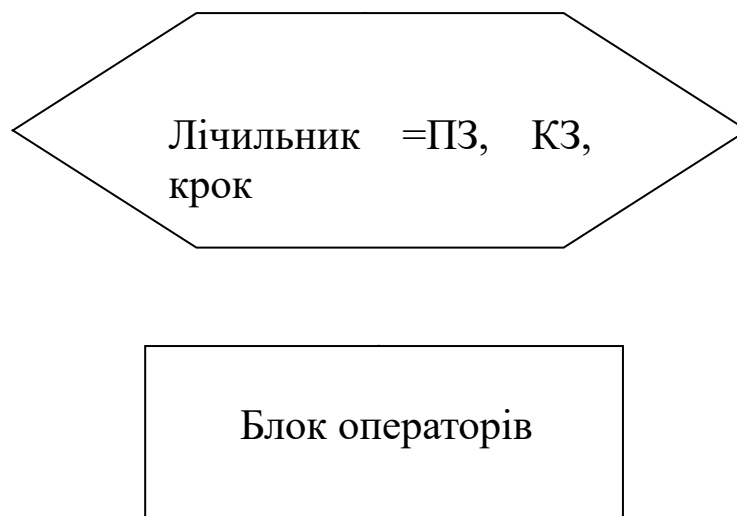
Знач_кроку – значення кроку, величина, що додається до лічильника після кожної ітерації. Вона може бути задана числом, змінною або математичним виразом. Додатне значення кроку використовується тоді, коли початкове значення лічильника менше кінцевого. Якщо початкове значення більше кінцевого, то слід застосовувати від'ємний крок.

Exit For – команда на достроковий (примусовий) вихід з циклу. У результаті її дії керування відразу передається першому

оператору нижче Next, незалежно від значення лічильника.

Next – повертає керування в рядок For.... Назву лічильника за цією командою вказувати не обов'язково. Якщо програмний блок містить декілька вкладених циклів, то за Next можна ставити кілька лічильників через кому.

Якщо в процесі роботи циклу крок додатний, то цикл буде завершений, коли лічильник стане більше кінцевого значення. Якщо в процесі роботи крок від'ємний, то керування піде із циклу після того, як лічильник стане менше свого кінцевого значення. У тому випадку, коли в програмі не зазначена одна з команд Next або For, а інша введена, при запуску буде видане повідомлення про помилку “For without Next” або “Next without For”. Зображення оператора For...Next у блок-схемі показано на рисунку 7.1.



ПЗ – початкове значення лічильника;

КЗ – кінцеве значення лічильника

Рисунок 7.1 – Зображення оператора For...Next на блок схемах

Приклад 7.1. Обчислити суму всіх цілих чисел від 1 до 10.

Розв'язання

```
S=0
For i = 1 To 10
  S = S + i
Next i
txtRez.Text = S
```

Робота програми. Програма повторює дію $S = S + i$ 10 разів, складаючи суму всіх цілих чисел від 1 до 10:

$$S = 1+2+3+4+5+6+7+8+9+10.$$

Лічильник циклу i після кожної ітерації збільшується на 1.

Сума накопичується в оперативній пам'яті змінної S .

При першій ітерації $S = S + i = 0 + 1 = 1$.

При другій $S = S + i = 1 + 2 = 3$.

У ході чергової ітерації програма додає до значення, що зберігається в S , поточне значення i , розраховує суму й записує її в комірку пам'яті, у якій розташована змінна S .

У математиці сума різних чисел записується за допомогою знака Σ . Сума виразів $(x-1)$ від 4 до n має такий вигляд:

$$U = \sum_{x=4}^n (x-1) .$$

Приклад 7.2 Розрахувати добуток кубів цілих чисел від 1 до n .

```
P = 1
For x = 1 To n
  P = P * x ^ 3
Next x
txtRez.Text = P
```

Робота програми. Змінній присвоюється значення 1, інакше VBA приймає, що P дорівнює 0 та в кожній ітерації чергове значення лічильника x буде множитися на 0 і загальний результат обчислень теж буде дорівнювати 0. Хід розрахунків аналогічний прикладу 7.1 з тою лише різницею, що відбувається не

додавання, а множення. У кожній ітерації циклу значення лічильника збільшується на 1, ця величина підноситься до куба і множить на поточне значення Р. Потім отриманий результат записується в комірку оперативної пам'яті, яка зберігає величину Р.

У математиці добуток чисел записується за допомогою знака П. Зокрема добуток кубів цілих чисел від 1 до n запишеться так:

$$P = \prod_1^n x^3.$$

Коли лічильник циклу змінюється із кроком 1, то використовувати крок циклу не потрібно й для програмування таких задач можна користуватися прикладами 7.1 і 7.2. В інших випадках числова послідовність змінюється з певним кроком, який слід указувати в рядку з For (приклад 7.3).

Приклад 7.3. Розрахувати

$$N = \frac{\sum x^2}{\prod (x+2)}, \text{ для } x = 1, 3, 5, 7, 9, 11.$$

```
P = 1
For x = 1 To 11 Step 2
  S = S + x^2
  P = P * (x + 2)
Next x
N = S / P
txtres.Text = N
```

Робота програми. У циклі розраховуються сума й добуток чисел, а потім програма обчислює приклад і виводить результат на екран. Числа, яким стає рівний лічильник x, у кожній ітерації змінюються із кроком 2, тому в першій ітерації x=1, у другій – x=3, у третій x=5 і т.п. до x=11.

Порядок виконання лабораторної роботи

- 1 Запустити програму Visual Basic for Applications.
- 2 Вивести на форми кнопки, надписи й текстові вікна, як того потребують завдання.
- 3 Присвоїти елементам відповідні властивості.

4 Провести розрахунки. Програма повинна містити введення вхідних даних, обчислення системи рівнянь та виведення результатів на екран.

Зміст звіту

У звіті повинно бути:

- 1 Визначена мета завдання.
- 2 Розв'язання завдання згідно з варіантом.
- 3 Наведено рисунки з елементами керування: форма із кнопками, текстові вікна, надписи.
- 4 Надано результати обчислень.
- 5 Висновки по лабораторній роботі.

Варіанти індивідуальних завдань до лабораторної роботи 7

Варіант	Завдання
1	Скласти алгоритм та програму табулювання функції $y=x^2+2x+5$ на відрізку $[1,2]$ з кроком 0,1
2	Скласти алгоритм та програму табулювання функції $y=3x^3-x^2$ на відрізку $[3,5]$ з кроком 0,5
3	Скласти алгоритм та програму знаходження суми чисел від N до M
4	Скласти алгоритм та програму розрахунку виразу $y = \sum_{i=1}^9 i^2 + \prod_{i=1}^9 i^3$
5	Знайти суму непарних чисел від 20 до 40. Скласти алгоритм та програму
6	Скласти алгоритм та програму розрахунку виразу $y = \sum_{i=10}^{15} i^2 + \prod_{x=1}^6 x^5$
7	Скласти алгоритм та програму розрахунку суми ряду $S=1/2+2/3+\dots+(n-1)/n$
8	Скласти алгоритм та програму розрахунку суми ряду $S=1+1/2^2+1/3^2+\dots+1/n^2$

Варіант	Завдання
9	Скласти алгоритм та програму розрахунку суми ряду $S_n=1+1/2+1/4+ \dots 1/2^n$
10	Скласти алгоритм та програму розрахунку суми ряду $X_n=1-1/3+1/5-\dots(-1)^{n-1}/(2n-1)$
11	Скласти алгоритм та програму розрахунку суми ряду $S=1+1/3^2+\dots+1/(2n-1)^2$
12	Скласти алгоритм та програму розрахунку суми ряду $S=x-x^2/2+x^3/3-x^4/4+\dots+(-1)^{n-1}\cdot x^n/n$
13	Скласти алгоритм та програму розрахунку добутку ряду $Y=n\cdot(n-1)\cdot(n-2)\cdot\dots\cdot(n-k+1)$
14	Скласти алгоритм та програму розрахунку суми добутків ряду $S=1\cdot 2+1\cdot 2\cdot 3+\dots+1\cdot 2\cdot 3\cdot \dots\cdot n$
15	Скласти алгоритм та програму розрахунку суми ряду $S=1\cdot x/1+x^2/1\cdot 2+\dots+x^n/1\cdot 2\cdot 3\cdot \dots\cdot n$

Питання до захисту лабораторної роботи 7

- 1 Які оператори циклу використовуються в VB?
- 2 Навести формат циклу For...Next і описати його дію.
- 3 Скласти програму розрахунків суми всіх цілих чисел від 10 до 20.
- 4 Скласти програму розрахунків добутку всіх цілих чисел від 1 до 10.

ЛАБОРАТОРНА РОБОТА 8

ОДНОВИМІРНІ МАСИВИ (ВЕКТОРИ) ДАНИХ

У VISUAL BASIC FOR APPLICATIONS

Мета роботи: ознайомитися з поняттям масиву даних і методами його обробки.

Завдання до лабораторної роботи

У даній лабораторній роботі потрібно обчислити усі приклади до введення та використання масивів (8.1-8.5) та скласти програму розрахунків математичних прикладів (варіанти завдань вказує викладач), що містить формування масиву та обробку його.

Короткі теоретичні положення

Масив даних – сукупність однотипних значень, об'єднана загальною назвою. Масиви бувають одновимірні (вектор), двовимірні (матриця) і тривимірні (група матриць). Кожний елемент масиву має свої координати. У векторах кожний елемент має порядковий номер, у двовимірних масивах позиція кожного елемента визначається двома координатами – номером рядка й номером стовпця. У тривимірних масивах до номера рядка й стовпця додається третій параметр – номер матриці.

У роботі з масивами можна виділити три основні завдання: формування масиву; обробка масиву; виведення результатів на екран.

Залежно від розмірності масиву й розв'язуваної задачі використовуються різні алгоритми, але принципи їх роботи однакові.

Формування масиву та оголошення масиву. Перш ніж сформувати масив, його необхідно оголосити. Оголошити масив – це значить дати вказівку комп'ютеру на резервування відповідної кількості комірок оперативної пам'яті для розміщення елементів масиву. Оголошення масиву, як і змінних, здійснюється за допомогою оператора Dim.

Формат оператора Dim:

$$\text{Dim Ім'я_масиву}(N) [\text{As тип}],$$

де N – кількість елементів в одновимірному масиві (векторі). Якщо масив має більше однієї розмірності, то в дужках вказується кількість елементів по кожній розмірності.

Наприклад, матриця C, у якій 4 рядки й 5 стовпців повинна оголошуватися як Dim C(1 to 4, 1 to 5) або Dim C(3, 4).

Тип – службове слово, що позначає тип даних, які будуть зберігатися в масиві, що оголошується. Наприклад, вектор A, який повинен зберігати 6 цілих чисел, можна оголосити так:

$$\text{Dim A}(5) \text{ As Integer.}$$

Якщо слово As і тип не зазначені, то програма вважає, що даний масив має тип Variant.

У програмі звернення до елемента масиву обов'язково супроводжується вказівкою його координат. У даній лабораторній роботі розглядаються одновимірні масиви (вектори), отже, для використання елементів вектора потрібно вказувати назву масиву й порядковий номер елемента. Номер записується в круглих дужках після назви масиву.

Наприклад, якщо масив B(1 to 5) містить п'ять елементів: B(1)=12, B(2)=5, B(3)=2, B(4)=11, B(5)=4, то сума другого й п'ятого елементів у програмі буде записана таким чином:

$$\text{Sum} = \text{B}(2) + \text{B}(5)$$

Одержуючи номер, програма звертається до відповідної комірки оперативної пам'яті, зчитує значення елемента й використовує його в розрахунках.

Оператор Dim слід указувати на початку програми й тільки 1 раз для кожного елемента, що оголошується. В одному операторі можна резервувати місце для декількох масивів і змінних, розділяючи їх комами. Наприклад:

```
Dim B(7) As Long, A(15) As Integer, i As Integer
```

Заповнення масиву – присвоєння кожному його елементу якого-небудь значення. Усі значення елементів мають той самий тип, зазначений при оголошенні масиву.

Розглянемо два найпоширеніші варіанти формування масиву.

1 Пряме присвоєвання. У цьому випадку вказують елемент із конкретним номером і присвоюють йому відповідне значення.

Приклад 8.1. Присвоїти п'яти елементам вектора A значення 5, 15, 4, 3 і 7.

```
Dim A(1 to 5) As Integer, i As Integer
Private Sub cmdArray_Click()
    A(1) = 5
    A(2) = 15
    A(3) = 4
    A(4) = 3
    A(5) = 7
    FOR i=1 TO 5
        txtRes.Text=txtRes.Text+Str(A(i))+vbNewLine
    Next i
End Sub
```

Даний варіант формування масиву зручний тільки в тих випадках, коли програма працює з тим самим набором даних. Якщо дані повинні змінюватися, то потрібно використовувати нижченаведений варіант формування.

2 *Уведення із клавіатури.* У цьому випадку після запуску програми користувач уводить значення кожного елемента самостійно за допомогою клавіатури. Програма задане число разів виводить на екран запит про введення наступного значення, а користувач уводить число із клавіатури. Для виведення запитів використовується функція `InputBox`.

У спрощеному вигляді її формат виглядає так: `InputBox("Текст")`, де "Текст" – це коментар, що пояснює мету запиту. Він повинен бути взятий у лапки.

Приклад 8.2. Увести значення вектора $A(1 \text{ to } 5)$ із клавіатури.

```
Dim A(1 to 5) As Integer, i As Integer
Private Sub CommandButton1_Click()
For i = 1 To 5
    A(i) = InputBox(Str(i) & " " & "елемент масиву")
    txtRes.Text = txtRes.Text + Str(A(i)) + vbNewLine
Next i
End Sub
```

Робота програми. Після клацання по кнопці «Введення елементів масиву» на формі, програма 5 разів виводить запит, показаний на рисунку 8.1. Користувач повинен ввести число в текстове вікно запиту й натиснути `Enter` (на клавіатурі) або клацнути мишкою по кнопці «Ок». Після цього, введене число автоматично буде присвоєне i -му елементу масиву. На першій ітерації – першому, на другій – другому й т.д.

Важливо відзначити, якщо число, яке вводиться через `InputBox`, присвоюється змінній, тип якої не оголошений, то це число буде присвоєно як текст і змінна буде вважатися текстовою. У таких випадках потрібно оголосити змінну із числовим типом або використовувати `InputBox` разом з функцією `Val`.

Наприклад, `X = Val(InputBox("Уведіть X"))`.

Перевагою введення із клавіатури є можливість забезпечення певного набору даних при кожному запуску програми.

Обробка й використання масиву – це аналіз значень його

елементів. Наприклад, пошук максимуму або мінімуму, розрахунки суми всіх елементів та ін. Використання масиву – це застосування значень його елементів для розрахунків яких-небудь величин або виконання інших дій.

Після того, як масив тим або іншим способом сформований, програма може працювати з ним. При цьому кожне звернення до елемента масиву повинне супроводжуватися вказівкою імені масиву й порядкового номера елемента.

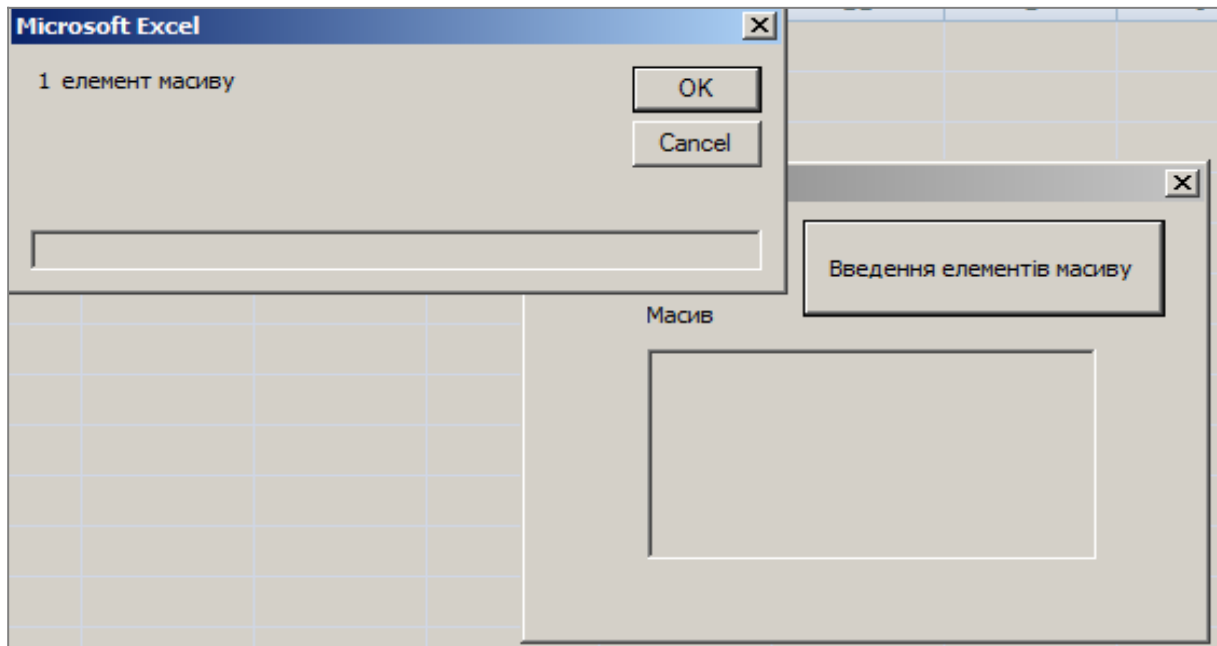


Рисунок 8.1

Приклад 8.3. Розрахувати середнє арифметичне вектора $A(1 \text{ to } 5)$, що містить п'ять елементів: $A(1)=1$, $A(2)=14$, $A(3)=5$, $A(4)=7$, $A(5)=9$.

```
Dim A(5) As Integer, i As Integer, s As Integer
Private Sub cmdCalculate_Click()
'Формування вектора
A(1)=1: A(2)=14: A(3)=5: A(4)=7: A(5)=9
' Розрахунок суми усіх елементів вектора
For i = 1 To 5
    S = S + A(i)
Next i
' Виведення середнього арифметичного
```

```
txtResult.Text = S / 5  
End Sub
```

Робота програми. Масив і всі допоміжні змінні оголошені за допомогою оператора Dim. Оголошення розміщене в розділі General. У процедурі cmdCalculate розміщено три блоки: формування вектора методом прямого присвоювання, розрахунки суми усіх елементів вектора й виведення на екран середнього арифметичного значення.

При аналізі даного алгоритму треба особливу увагу звернути на розрахунок суми. Вона накопичується в змінній s. У кожній ітерації до поточного значення s додається черговий (і-й) елемент масиву. Зміна нумерації елементів досягається тим, що лічильник циклу поставлений як покажчик номера елемента. Лічильник змінюється від 1 до 5 із кроком +1, отже, його значення збігаються з номерами елементів вектора – 1, 2, 3, 4 і 5.

Перша ітерація (i=1): $s = s + A(i) = 0 + A(1) = 0 + 1 = 1$.

Друга ітерація (i=2): $s = s + A(i) = 1 + A(2) = 1 + 14 = 15$.

Й так далі до п'ятої ітерації (i=5).

Виведення вектора на екран. Алгоритм виведення масиву в текстове вікно залежить від розмірності самого масиву й тієї форми, у якій він повинен бути поданий на екрані. Алгоритм виведення вектора в текстове вікно у вигляді стовпця чисел розглянуто в прикладі 8.4.

Якщо масив виводиться в текстове вікно й містить більш одного рядка, то в ході настроювання текстового вікна потрібно перемкнути його властивість Multiline_у положення True. За замовчуванням значення цієї властивості False, що забороняє виводити більш одного рядка в текстове вікно.

Приклад 8.4. Вивести вектор A з п'ятьма елементами в текстове вікно txtVector у вигляді стовпця чисел.

```
For i = 1 To 5  
    txtVector.Text = txtVector.Text + Str(A(i)) + vbNewLine  
Next i  
End Sub
```

Робота програми. Якщо властивість `MultiLine` текстового вікна `txtVector` дорівнює `True`, то алгоритм виведе значення елементів вектора в один стовпець. Елементи виводяться один по одному, відповідно до зміни лічильника циклу `i`.

У кожній ітерації програма переводить чергове значення елемента в текстову форму (`Str(A(i))`), додає його до вмісту текстового вікна (`txtVector.Text + Str(A(i))`), переводить курсор на наступний рядок (`vbNewLine`) і виводить отриманий текст у текстове вікно.

Інформація у вікні накопичується як сума текстових фрагментів. Причому функція `vbNewLine` записує в текст дві команди – перейти на наступний рядок + поставити позицію виводу в початок рядка. Якщо `vbNewLine` не використовувати, то вектор буде виведений в один рядок.

Приклад 8.5. Сформувати вектор `A` з елементами 4, 6, 7, 9, 6, 8 методом прямого присвоювання й вивести на екран. Для кожного елемента вектора розрахувати значення $Y = A^2$.

```
Dim A(1 to 6) As Integer, i As Integer, Y As Integer
Private Sub cmdVect_Click()
' Формуємо вектор
A(1) = 4: A(2) = 6: A(3) = 7: A(4) = 9: A(5) = 6: A(6) = 8
End Sub
```

```
Private Sub cmdPrint_Click()
' Виведення вектор в текстове вікно як стовпець
For i = 1 To 6
txtVector.Text = txtVector.Text + Str(A(i)) + vbNewLine
Next i
End Sub
```

```
Private Sub cmdResult_Click()
' Розрахунок функції та виведення результату
For i = 1 To 6
Y = A(i)^2
txtResult.Text = txtResult.Text + Str(Y) + vbNewLine
Next i
End Sub
```

```
Private Sub cmdClear_Click()  
txtVector.Text = " "  
txtResult.Text = " "  
End Sub
```

```
Private Sub cmdExit_Click()  
End  
End Sub
```

Робота програми. Рекомендований зовнішній вигляд форми на дану роботу надано на рисунку 8.2.

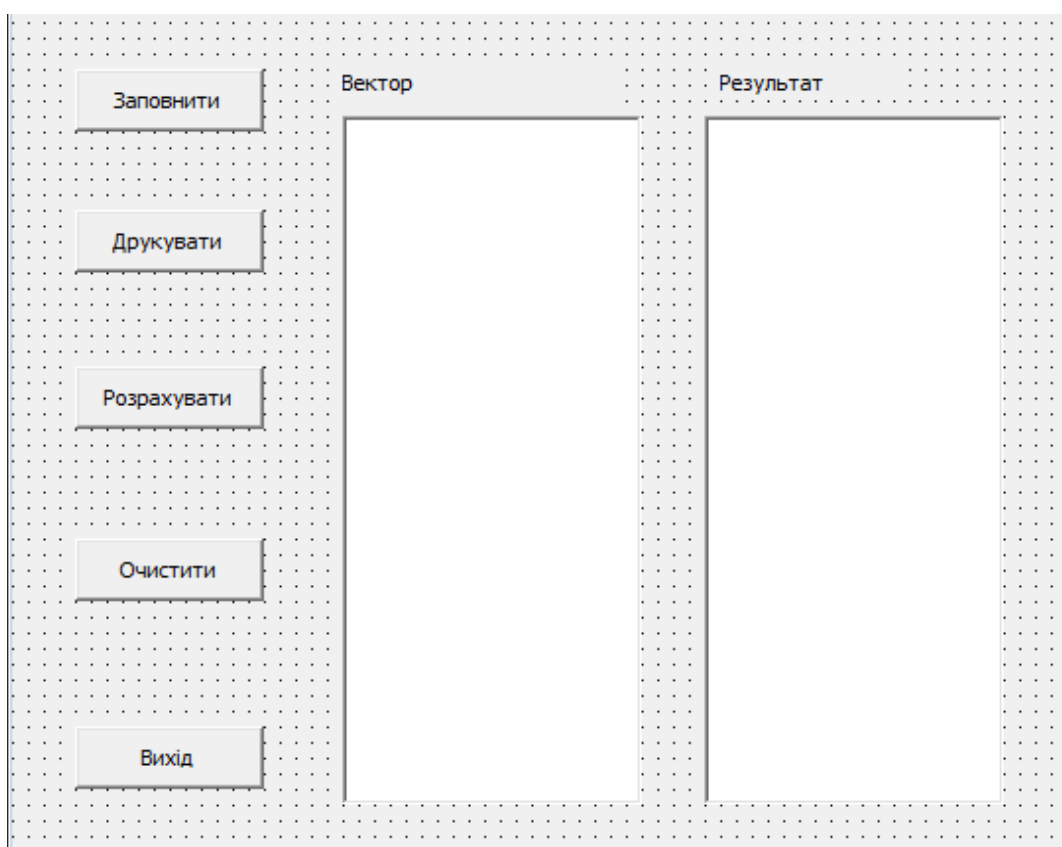


Рисунок 8.2

Формування вектора й обчислення функції для кожного з його елементів подано в програмі, наведеній вище. Після натискання на кнопку «Заповнити» спрацьовує процедура cmdVector, у якій відбувається формування вектора A. Клацання мишкою по кнопці – «Друкувати» запускає процедуру cmdPrint, яка виводить елементи сформованого вектора в текстове вікно

txtVector у вигляді стовпця. Після запуску програми потрібно натискати на кнопки в послідовності: «Заповнити»; «Друкувати»; «Розрахувати»; «Очистити», при необхідності, «Вихід».

Кнопка «Розрахувати» відповідає процедурі cmdResult. В цій процедурі черговий елемент вектора зводиться у квадрат, результат присвоюється змінній Y і виводиться в текстове вікно txtResult.

Властивість MultiLine текстових вікон встановлюється True.

Порядок виконання лабораторної роботи

- 1 Запустити програму Visual Basic for Applications.
- 2 Вивести на форми кнопки, надписи й текстові вікна, як того потребують приклади (8.1-8.5).
- 3 Присвоїти елементам властивості.
- 4 Увести елементи масивів.
- 5 Провести розрахунки, якщо це необхідно згідно з варіантом, та вивести результати на екран.

Зміст звіту

У звіті повинно бути:

- 1 Визначена мета завдання.
- 2 Наведено рисунки з елементами керування: форма із кнопками, текстове вікно, надписи.
- 3 Перелічені основні властивості використаних об'єктів керування.
- 4 Висновки по лабораторній роботі.

Варіанти індивідуальних завдань до лабораторної роботи 8

Варіант	Завдання
1	Шляхом уведення із клавіатури сформувати вектор X з 8 елементів. Вивести елементи вектора на екран. Обчислити середнє геометричне елементів масиву
2	Методом прямого присвоювання сформувати вектор Y з 9 елементів. Вивести елементи вектора на екран. Обчислити середнє арифметичне елементів масиву
3	Шляхом уведення із клавіатури сформувати вектор Z з 6 елементів. Вивести елементи вектора на екран.

	Знайти суму елементів вектора
4	Методом прямого присвоювання сформувати вектор А з 10 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран добуток елементів вектора
5	Шляхом уведення із клавіатури сформувати цілочисловий вектор В з 5 елементів. Вивести елементи вектора на екран. Визначити, чи кратна сума його елементів 7

Варіант	Завдання
6	Методом прямого присвоювання сформувати цілочисловий вектор С з 8 елементів. Вивести елементи вектора на екран. Обчислити середнє геометричне елементів, кратних 3
7	Шляхом уведення із клавіатури сформувати вектор D з 11 елементів. Вивести на екран елементи вектора з парними індексами
8	Методом прямого присвоювання сформувати вектор E з 7 елементів. Вивести елементи вектора на екран. Вивести на екран елементи вектора з непарними індексами
9	Шляхом уведення із клавіатури сформувати вектор F з 8 елементів. Знайти й вивести на екран суму елементів, що кратні 5
10	Методом прямого присвоювання сформувати вектор G з 25 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран суму п'ятого та сьомого елементів вектора
11	Шляхом уведення із клавіатури сформувати вектор H з 10 елементів. Вивести елементи вектора на екран. Вивести на екран суму першого і другого елемента вектора
12	Методом прямого присвоювання сформувати вектор J з 8 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран суму та добуток елементів вектора

13	Шляхом уведення із клавіатури сформувати вектор К з 7 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран добуток непарних елементів вектора
14	Методом прямого присвоювання сформувати вектор L з 10 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран добуток парних елементів вектора
15	Шляхом уведення із клавіатури сформувати вектор M з 9 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран п'ятий елемент вектора

Питання до захисту лабораторної роботи 8

- 1 Що таке масив даних?
- 2 Чим відрізняються одновимірні, двовимірні й тривимірні масиви?
- 3 Для чого проводять оголошення масиву?
- 4 Навести формат оператора Dim і описати його роботу.
- 5 У якому місці програми рекомендується розміщати рядок оголошення масиву?
- 6 Як працює алгоритм заповнення масиву методом прямого присвоювання? Навести приклад.
- 7 Навести формат функції InputBox.
- 8 Як працює алгоритм заповнення масиву із клавіатури?
- 9 Чим відрізняється обробка й використання масиву?
- 10 Навести алгоритм розрахунків суми всіх елементів вектора й описати його роботу.
- 11 Для чого використовується властивість текстового вікна MultiLine?
- 12 Для чого використовується функція vbNewLine?

ЛАБОРАТОРНА РОБОТА 9
ВИЗНАЧЕННЯ МАКСИМАЛЬНОГО
(МІНІМАЛЬНОГО) ЕЛЕМЕНТА ВЕКТОРА
У VISUAL BASIC FOR APPLICATIONS

Мета роботи: ознайомитися з алгоритмами пошуку екстремумів вектора.

Завдання до лабораторної роботи

У даній лабораторній роботі потрібно обчислити усі наведені приклади до введення та використання масивів (9.1-9.2). У ході даної лабораторної роботи потрібно скласти програму, яка формує й виводить на екран вектор, а потім визначає його максимум або мінімум згідно з варіантом, що вказує викладач.

Короткі теоретичні положення

Алгоритми пошуку максимуму й мінімуму є найбільш відомими задачами обробки масивів. Вони засновані на порівнянні поточного елемента вектора з початковим значенням змінної Max або Min.

Пошук максимуму: спочатку змінній присвоюється початкове значення, як правило, ним є значення першого елемента вектора, потім кожний елемент вектора порівнюють із змінною Max.

Якщо при пошуці максимуму елемент більше Max, то Max присвоюється значення даного елемента й подальше порівняння ведуть із новим значенням Max. Після того, як будуть переглянуті всі елементи вектора, значення, що залишилося в змінній Max, й буде рівне максимальному.

Пошук мінімуму ведеться аналогічно, з тією лише різницею, що в ході перевірки програма шукає елементи, значення яких менші Min.

Приклад 9.1. Знайти максимальний елемент вектора X(1 to 20).

```
Max = X(1)           '1
For i = 2 To 20      '2
  If X(i) > Max Then Max = X(i)  '3
Next i
txtResult.Text = Max '4
```

Робота програми. Змінній Max присвоюється значення першого елемента вектора X(1). Потім у циклі перебираються всі елементи вектора, починаючи із другого. Якщо $X(i) > \text{Max}$, то змінній Max присвоюється значення цього елемента ($\text{Max} = X(i)$) і подальший аналіз іде вже щодо нового значення Max. Якщо умова $X(i) > \text{Max}$ не виконується, то величина змінної Max залишиться незмінною. Пошук мінімуму працює аналогічно, тільки перевіряють протилежну умову:

$$\text{If } X(i) < \text{Min} \text{ Then Min} = X(i)$$

Приклад 9.2. Шляхом уведення із клавіатури сформувати вектор X(1 to 6), вивести його на екран і знайти мінімальне значення.

```
Dim X(1 to 6) As Integer, i As Integer, Min As Integer
Private Sub cmdVector_Click()
```

```

'Формування вектора
For i = 1 To 6
  X(i) = InputBox("Введіть елемент вектора")
Next i
End Sub

Private Sub cmdPrint_Click()
'Виведення вектора в текстове вікно
For i = 1 To 6
  txtPrint.Text = txtPrint.Text + Str(X(i)) + vbCrLf
Next i
End Sub

Private Sub cmdMin_Click()
'Пошук мінімуму
Min = X(1)
For i = 2 To 6
  If X(i) < Min Then Min = X(i)
Next i
txtResult.Text = Min
End Sub

Private Sub cmdExit_Click()
  End
End Sub

```

Робота програми. У процедурі
Private Sub cmdVector_Click()
...
End Sub

реалізований алгоритм уведення значень елементів вектора із клавіатури.

Виведення сформованого вектора в текстове вікно відбувається в процедурі

```

Private Sub cmdPrint_Click()
...
End Sub.

```

Пошук мінімуму й виведення результату на екран реалізовано в процедурі

```

Private Sub cmdMin_Click()

```

...

End Sub

Натискати на кнопки, розташовані на формі, потрібно в такій послідовності: “Заповнити”, “Друкувати”, “Min”. Це пов'язано з тим, що кожна наступна процедура використовує у своїй роботі результати попередніх процедур.

Порядок виконання лабораторної роботи

- 1 Запустити програму Visual Basic for Applications.
- 2 Вивести на форми кнопки, надписи й текстові вікна, як того потребує завдання, (згідно з варіантом, указаним викладачем).
- 3 Увести елементи вектора.
- 4 Провести розрахунки згідно з варіантом та вивести результати на екран.

Зміст звіту

У звіті повинно бути:

- 1 Визначена мета завдання.
- 2 Наведено рисунки з елементами керування: форма із кнопками, текстове вікно, етикетки.
- 3 Висновки по лабораторній роботі.

Варіанти індивідуальних завдань до лабораторної роботи 9

Варіант	Завдання
1	Шляхом уведення із клавіатури сформувати вектор X з 10 елементів. Вивести парні елементи вектора на екран. Знайти й вивести на екран максимальний елемент вектора
2	Методом прямого присвоювання сформувати вектор Y з 20 елементів. Вивести непарні елементи вектора на екран. Знайти й вивести на екран максимальний елемент вектора
3	Шляхом уведення із клавіатури сформувати вектор Z з 8 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран мінімальний елемент вектора
4	Методом прямого присвоювання сформувати вектор A

	з 15 елементів. Вивести додатні елементи вектора на екран. Знайти й вивести на екран максимальний елемент вектора
5	Шляхом уведення із клавіатури сформувати вектор В з 9 елементів. Вивести від'ємні елементи вектора на екран. Знайти й вивести на екран максимальний елемент вектора
6	Методом прямого присвоювання сформувати вектор С з 17 елементів. Вивести елементи вектора на екран у зворотному порядку. Знайти й вивести на екран мінімальний елемент вектора
7	Шляхом уведення із клавіатури сформувати вектор D з 11 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран максимальний та мінімальний елементи вектора
Варіант	Завдання
8	Методом прямого присвоювання сформувати вектор Е з 20 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран максимальний елемент вектора та його індекс
9	Шляхом уведення із клавіатури сформувати вектор F з 8 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран п'ятий та максимальний елемент вектора
10	Методом прямого присвоювання сформувати вектор G з 25 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран максимальний елемент вектора та його індекс
11	Шляхом уведення із клавіатури сформувати вектор H з 10 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран мінімальний елемент вектора та його індекс
12	Методом прямого присвоювання сформувати вектор J з 22 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран мінімальний елемент вектора та його індекс
13	Шляхом уведення із клавіатури сформувати вектор K з

	7 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран мінімальний елемент вектора та його індекс
14	Методом прямого присвоювання сформувати вектор L з 8 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран максимальний елемент вектора та його індекс
15	Шляхом уведення із клавіатури сформувати вектор M з 9 елементів. Вивести елементи вектора на екран. Знайти й вивести на екран максимальний елемент вектора та його індекс

Питання до захисту лабораторної роботи 9

- 1 Що таке вектор даних?
- 2 Як працює алгоритм пошуку максимуму вектора?
- 3 Як працює алгоритм пошуку мінімуму вектора?
- 4 Для чого використовується властивість текстового вікна MultiLine?

СПИСОК ЛІТЕРАТУРИ

- 1 Гарнаев А.Ю. Microsoft Excel 2002: Розробка приложень. – СПб.: ВHV-Санкт-Петербург, 2002. – 763 с.
- 2 Гарнаев А.Ю. Самоучитель VBA. – 2-е изд. СПб.: ВHV-Петербург, 2004. – 540 с.
- 3 Гарнаев А.Ю. VBA: в подлиннике. СПб.: ВHV-Санкт-Петербург, 2005. – 848 с.
- 4 Титаренко Г. Visual Basic 6.0. – К. : ВHV, 2002. – 447 с.
- 5 Створення програмних додатків в середовищі MS Office Visual Basic for Application (для самостійного вивчення): Навч. посібник / Укладачі Ш.Ф. Григоришин, С.В. Косяченко, Л.В. Кульбаба. – К.: Дакор: КНТ, 2007. – 208 с.

6 Карпов Б.И. Специальный справочник. VBA (Visual Basic for Applications). Скриптовый язык приложений Microsoft Office. – СПб.: Питер, 2002. - 416 с.

7 Кузьменко В.Г. Программирование на VBA 2002. – М.: Бином-Пресс, 2003. – 880 с.

8 Уокенбах Д. Профессиональное программирование на VBA в Excel 2002. – М.: Вильямс, 2003. – 784 с.

