

ФАКУЛЬТЕТ АВТОМАТИКИ, ТЕЛЕМЕХАНІКИ ТА ЗВ'ЯЗКУ

Кафедра обчислювальної техніки і систем управління

МЕТОДИЧНІ ВКАЗІВКИ

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

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

Харків – 2014

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри обчислювальної техніки та систем управління 26 лютого 2013 р., протокол № 8.

Рекомендуються для студентів будівельного та механічного факультету для спеціальностей 6.060101 «Будівництво», 6.070108 «Залізничні споруди та колійне господарство», 6.050503 «Машинобудування», 6.050601 «Теплоенергетика» всіх форм навчання.

Укладачі:

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

Рецензент

проф. С.В. Лістровий

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт по Visual Basic for Applications
з дисциплін
*«ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ»,
«ОСНОВИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ТА ПРОГРАМУВАННЯ»*

Відповідальний за випуск Піскачова І.В.

Редактор Ібрагімова Н.В.

Підписано до друку 03.04.13 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 1,00. Тираж 50. Замовлення №

Видавець та виготовлювач Українська державна академія залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

Міністерство транспорту та зв'язку України

Українська державна академія
Залізничного транспорту

Методичні вказівки
до лабораторних робіт по Visual Basic for Applications
з дисциплін «Обчислювальна техніка та програмування»,
«Основи інформаційних технологій та програмування»
для студентів будівельного та механічного факультету
Частина 3

Харків 2013

Методичні вказівки розглянуто і рекомендовано до друку на засіданні кафедри обчислювальної техніки та систем управління 26 лютого 2013 р., протокол № 8.

Рекомендуються для студентів будівельного та механічного факультету для спеціальностей 6.060101 «Будівництво», 6.070108 «Залізничні споруди та колійне господарство», 6.050503 «Машинобудування», 6.050601 «Теплоенергетика» всіх форм навчання.

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

Рецензент
проф. С.В. Лістровий

ЗМІСТ

Вступ.....	4
..	4
Лабораторна робота 10. Методи сортування масивів.....	5
Лабораторна робота 11. Робота з листами, комірками, діапазонами комірок Excel за допомогою VBA.....	13
Список літератури.....	22

ВСТУП

Мова програмування Visual Basic for Applications (VBA) застосовується для розширення функціональних можливостей Microsoft Office, у якому вона використовується, і є спрощеною версією Visual Basic, імовірно, найпопулярнішого засобу розробки. В VBA достатньо вивчити одне середовище програмування, щоб скласти різноманітні додатки.

Близько 40 компаній-виробників програмного забезпечення (Autodesk, Adobe, PeopleSoft, Baan, SAP, Solomon Software та ін.) уклали з корпорацією Microsoft угоду про включення VBA в їх продукти (AutoCAD, SolidWorks, CorelDRAW, WordPerfect, ESRI ArcGIS, 1С та ін.).

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 2 і присвячений вивченню сортування масивів і роботи з деякими об'єктами в середовищі VBA. Він допоможе студентам надалі використовувати отримані навички в професійній діяльності.

Лабораторна робота 10

МЕТОДИ СОРТУВАННЯ МАСИВІВ

Мета роботи: ознайомитися з деякими методами сортування масивів.

Завдання до лабораторної роботи

У даній лабораторній роботі потрібно відсортувати масив згідно з індивідуальним завданням.

Короткі теоретичні положення

Сортування – це загальна задача багатьох комп'ютерних додатків. Практично будь-який список цінніший, коли він відсортований за яким-небудь певним принципом.

Нехай є послідовність $a_0, a_1 \dots a_n$ і функція порівняння, що на будь-яких двох елементах послідовності приймає одне із трьох значень: менше, більше або дорівнює. Завдання сортування полягає в перестановці членів послідовності таким чином, щоб виконувалася умова $a_i \leq a_{i+1}$ для всіх i від 0 до n .

Для того щоб обґрунтовано зробити такий вибір, розглянемо параметри, по яких буде вироблятися оцінка алгоритмів.

Термін сортування – основний параметр, що характеризує швидкодію алгоритму.

Деякі алгоритми вимагає виділення додаткової пам'яті під тимчасове зберігання даних. При оцінці використовуваної **пам'яті** не враховується місце, що займає вихідний масив, і незалежні від вхідної послідовності витрати, наприклад, на зберігання коду програми.

Стійкість – сортування не змінює взаємного розташування рівних елементів.

Природність поводження – ефективність методу при обробці вже відсортованих або частково відсортованих даних.

Ще одною важливою властивістю алгоритму є його сфера застосування:

- внутрішнє сортування, що оперує з масивами, які цілком містяться в оперативній пам'яті з довільним доступом до будь-

якої комірки. Дані звичайно сортуються на тім же місці без додаткових витрат;

- зовнішні сортування впорядковують інформацію, розташовану на зовнішніх носіях. Це накладає деякі додаткові обмеження на алгоритм: доступ до носія здійснюється послідовно: у кожний момент часу можна записати тільки елемент, що впливає за поточним, об'єм даних не дозволяє їм розміститися в ОЗУ. Крім того, доступ до даних на носії здійснюється набагато повільніше, ніж операції з оперативною пам'яттю.

Методи внутрішнього сортування

У загальній постановці задача ставиться в такий спосіб. Є послідовність однотипних записів, одне з полів яких обране в якості ключового (ключ сортування). Тип даних ключа повинен включати операції порівняння ("=", ">", "<", ">=" і "<=").

Задачею сортування є перетворення вихідної послідовності в послідовність, що містить ті самі записи, але в порядку зростання (або зменшення) значень ключа.

Природною умовою, пропонованою до будь-якого методу внутрішнього сортування, є те, що ці методи не повинні вимагати додаткової пам'яті: всі перестановки з метою впорядкування елементів масиву повинні вироблятися в межах того самого масиву. Помітимо, що оскільки сортування засноване тільки на значеннях ключа й ніяк не стосується оставшихся полів записів, можна говорити про сортування масивів ключів.

Обмінне сортування (сортування методом пухирця)

Обмінне сортування – це алгоритм, призначений для сортування списків, які вже перебувають у майже впорядкованому стані. Якщо ця умова виконана, то алгоритм виконується дуже швидко.

Просте обмінне сортування для масиву $a(1), a(2), \dots, a(n)$ працює в такий спосіб. Починаючи з кінця масиву порівнюються два сусідніх елементи ($a(n)$ і $a(n-1)$). Якщо виконується умова $a(n-1) > a(n)$, то значення елементів міняються місцями. Процес триває для $a(n-1)$ і $a(n-2)$ і т. д., поки не буде зроблено порівняння $a(2)$ і $a(1)$. Зрозуміло, що після цього на місці $a(1)$ опиниться елемент масиву з найменшим значенням. На другому кроці

процес повторюється, але останніми рівняються $a(3)$ і $a(2)$. І так далі. На останньому кроці будуть порівнюватися тільки поточні значення $a(n)$ і $a(n-1)$. Зрозуміла аналогія з пухирцем, оскільки найменші елементи (самі "легкі") поступово "спливають" до верхньої границі масиву. Приклад сортування методом пухирця показаний на рисунку 10.1.

Початковий стан масиву	8 23 5 65 44 33 1 6
Крок 1	
	8 23 5 65 44 33 1 6
	8 23 5 65 44 1 33 6
	8 23 5 65 1 44 33 6
	8 23 5 1 65 44 33 6
	8 23 1 5 65 44 33 6
	8 1 23 5 65 44 33 6
	1 8 23 5 65 44 33 6
Крок 2	
	1 8 23 5 65 44 6 33
	1 8 23 5 65 6 44 33
	1 8 23 5 6 65 44 33
	1 8 23 5 6 65 44 33
	1 8 5 23 6 65 44 33
	1 5 8 23 6 65 44 33
Крок 3	
	1 5 8 23 6 65 33 44
	1 5 8 23 6 33 65 44
	1 5 8 23 6 33 65 44
	1 5 8 6 23 33 65 44
	1 5 6 8 23 33 65 44
Крок 4	
	1 5 6 8 23 33 44 65
	1 5 6 8 23 33 44 65
	1 5 6 8 23 33 44 65
	1 5 6 8 23 33 44 65
Крок 5	
	1 5 6 8 23 33 44 65
	1 5 6 8 23 33 44 65
	1 5 6 8 23 33 44 65
Крок 6	
	1 5 6 8 23 33 44 65
	1 5 6 8 23 33 44 65
Крок 7	
	1 5 6 8 23 33 44 65

Рисунок 10.1 – Приклад сортування методом пухирця

Метод пухирця допускає три простих удосконалення. По-перше, як показує рисунок 10.1, на чотирьох останніх кроках розташування значень елементів не змінювалося (масив виявився

вже впорядкованим). Тому якщо на деякому кроці не було зроблено жодного обміну, то виконання алгоритму можна припинити. По-друге, можна запам'ятовувати найменше значення індексу масиву, для якого на поточному кроці виконувалися перестановки.

Приклад програми для сортування методом пухирця:

```
Option Explicit
Dim n As Integer
Dim a(1 To 10) As Variant, i As Integer, j As Integer
Dim p As Variant, l As Integer

Private Sub CommandButton1_Click()
    n = Val(txtn.Text)
    For i = 1 To n
        a(i) = Int(Rnd * 100 - 50)
        txtvector.Text = txtvector.Text + Str(a(i))
    Next i
    For i = 1 To n - 1
        For j = n To i + 1 Step -1
            If a(j - 1) > a(j) Then
                p = a(j - 1)
                a(j - 1) = a(j)
                a(j) = p
            End If
        For l = 1 To n
            txtsort.Text = txtsort.Text + Str(a(l)) + "  "
        Next l
        txtsort.Text = txtsort.Text + vbCrLf
    Next j
    Next i
End Sub
```

Приклад оформлення інтерфейсу користувача у VBA наведено на рисунку 10.2.

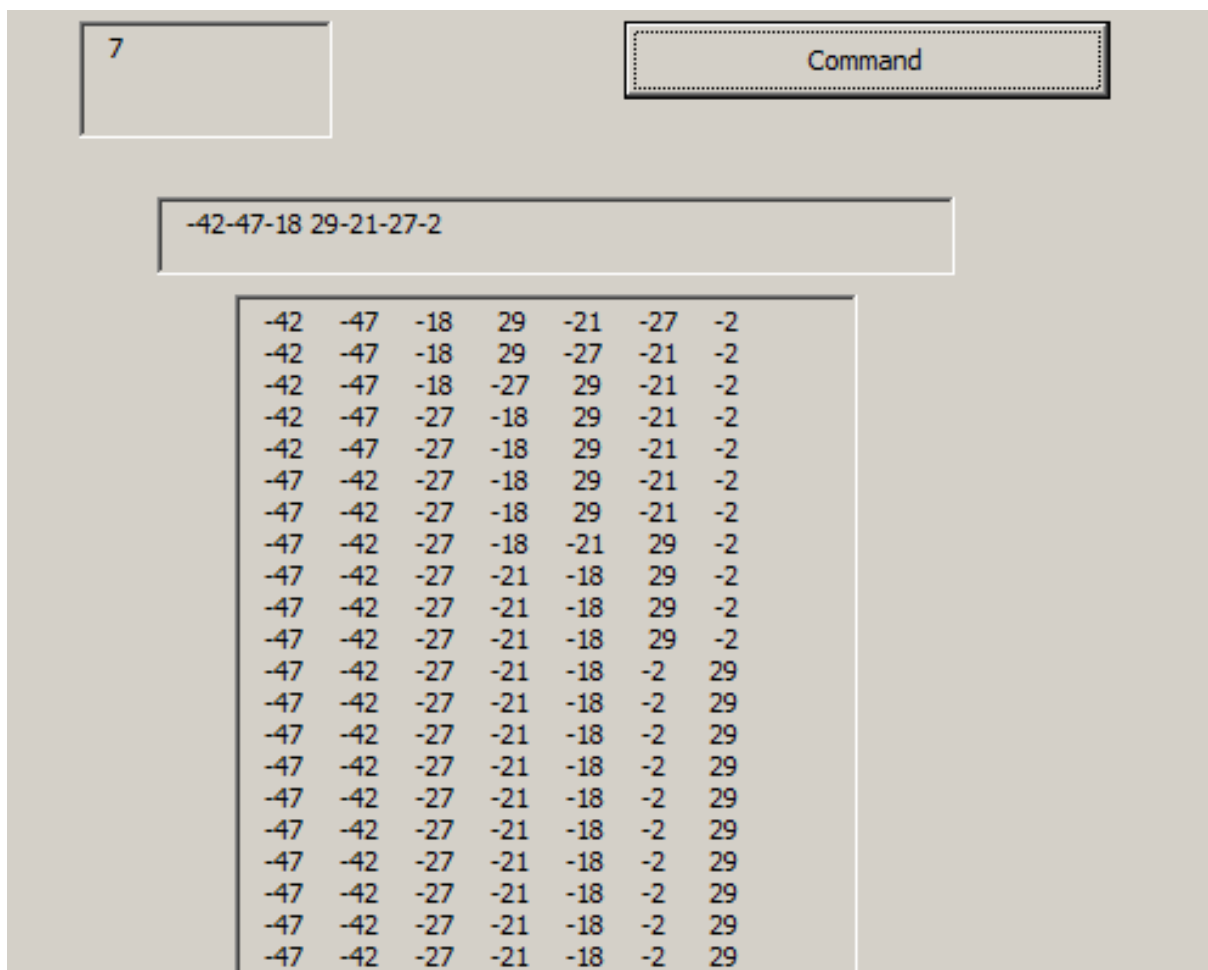


Рисунок 10.2 – Результат сортування масиву з 7 елементів методом пухирця

Сортування вибором

Сортування вибором – це простий алгоритм. Його задача – шукати найменший елемент масиву з n елементів, що потім міняється місцями з елементом з початку списку. Потім шукається найменший з елементів, що залишилися, і міняється місцями із другим елементом. Процес триває доти, поки всі елементи не займуть своє кінцеве положення.

2 4 5 3 10 8 1
 $a(1)$ $a(2)$ $a(3)$ $a(4)$ $a(5)$ $a(6)$ $a(7)$

Визначається найменший елемент $a(7)=1$.

Міняється місцями з першим

1 4 5 3 10 8 2

В елементах, що залишилися, знову шукається мінімальний і міняється із другим

```

1    2    5    3    10  8    4
і т. д.
1    2    3    5    10  8    4
1    2    3    4    10  8    5
1    2    3    4    5    8    10
1    2    3    4    5    8    10

```

Усього проходів n-1.

Для реалізації цього алгоритму необхідні вкладені цикли For...Next. Зовнішній цикл (по i) призначений для послідовного фіксування елементів масиву, внутрішній (по j) - здійснює пошук мінімуму (максимуму) і його позиції. Після виходу із внутрішнього циклу потрібна перестановка елементів. Останній елемент у зовнішньому циклі не розглядається: він сам стане на своє місце.

Option Explicit

Dim m As Integer

Dim a(1 To 10) As Variant, i As Integer, j As Integer

Dim min As Variant, l As Integer, kmin As Variant

Private Sub CommandButton2_Click()

 m = Val(txtm.Text)

 For i = 1 To m

 a(i) = Int(Rnd * 100 - 50)

 txtvector1.Text = txtvector1.Text + Str(a(i))

 Next i

 For i = 1 To m - 1

 min = a(i)

 kmin = i

 For j = i + 1 To m

 If a(j) < min Then

 min = a(j)

 kmin = j

 End If

 Next j

 a(kmin) = a(i)

 a(i) = min

```

For l = 1 To m
    txtsort1.Text = txtsort1.Text + Str(a(l)) + " "
Next l
    txtsort1.Text = txtsort1.Text + vbCrLf
Next i

End Sub

```

При пошуку i -го найменшого елемента алгоритм повинен перевірити кожний з $N-I$, що залишилися. Сортування вибором працює досить добре зі списками, де елементи розташовані випадково або в прямому порядку, але для обернено сортованих списків продуктивність цього алгоритму набагато гірша. Для пошуку мінімального елемента списку сортування вибором виконує таку послідовність операторів:

```

If a(j) < Min Then
    Min = a(j)
    k_min = j
End If

```

Якщо список відсортований у зворотному порядку, умова $a(j) < \text{Min}$ виконується більшу частину часу. Під час першого проходження через список елементів вона буде істинною для всіх елементів, тому що кожний елемент менший, ніж попередній. Програма виконує порівняння багато разів, що призводить до деякого вповільнення роботи алгоритму.

Це не найшвидший алгоритм, але він дуже простий, а також швидко сортує невеликі списки.

Завдання на лабораторну роботу 10

- 1 Побудувати алгоритм обмінного сортування (методу пухирця).
- 2 Відсортувати методом пухирця масив випадково заданих чисел.
- 3 Побудувати алгоритм сортування методом вибору.
- 4 Відсортувати методом вибору масив випадково заданих чисел.

Порядок виконання лабораторної роботи

- 1 Запустити програму Visual Basic for Applications.
- 2 Вивести на форми кнопки, надписи й текстові вікна, як того потребує завдання.
- 3 Присвоїти елементам відповідні властивості.
- 4 Провести сортування методом пухирця та вибору і вивести результати на екран.

Зміст звіту

У звіті має бути:

- 1) визначена мета завдання;
- 2) розв'язання завдання згідно з варіантом;
- 3) наведено рисунки з елементами керування: форма із кнопками, текстові вікна, написи;
- 4) надано результати сортування;
- 5) висновки з лабораторної роботи.

Варіанти індивідуальних завдань до лабораторної роботи 10

Варіант	Завдання
1	Змінити процедуру сортування методом пухирця так, щоб сортування відбувалось за зменшенням елементів
2	Відсортувати парні елементи масиву за допомогою простого вибору
3	Змінити процедуру сортування методом вибору так, щоб сортування відбувалось за зменшенням елементів
4	Відсортувати непарні елементи масиву за допомогою методу пухирця
5	Відсортувати за допомогою методу вибору елементи масиву, що мають непарні індекси
6	Впорядкувати елементи масиву, що мають непарні індекси, у зростаючому порядку, а парні – у порядку, що зменшується
7	У масиві $X(N)$ кожний елемент дорівнює 0, 1 або 2. Переставити елементи масиву так, щоб спочатку розташовувалися всі одиниці, потім всі двійки й, нарешті, всі нулі (додаткового масиву не заводити)

8	Із двох упорядкованих одномірних масивів (довжини K і N) сформууйте одномірний масив розміром K+N, упорядкований у зворотний бік
9	Із двох упорядкованих одномірних масивів (довжини K і N) сформууйте одномірний масив розміром K+N, упорядкований так само, як вихідні масиви
10	Із двох упорядкованих одномірних масивів (довжини K і N) сформууйте одномірний масив розміром K+N, упорядкований у зворотний бік.

Лабораторна робота 11

РОБОТА З ЛИСТАМИ, КОМІРКАМИ, ДІАПАЗОНАМИ КОМІРОК EXCEL ЗА ДОПОМОГОЮ VBA

Мета роботи: ознайомитися з об'єктами Sheet, Cell, Range, Application, властивостями, методами і подіями цих об'єктів.

Завдання до лабораторної роботи

У даній лабораторній роботі потрібно зробити завдання 1 та 2, відповісти на тестові питання.

Короткі теоретичні положення

Об'єкт - основний елемент VBA Excel. В VBA об'єктами є робоча книга, робочий лист, комірка, діапазон комірок та ін.

Приклади об'єктів:

Sheet – лист Excel;

Cell – комірка;

Range – діапазон комірок;

Application – додаток;

UserForm – користувальницька форма.

Доступ до об'єкта можливий через його методи й властивості.

Методи. Над об'єктами можна робити різні дії. Дії, що здійснюються над об'єктами, мають назву методи. Наприклад,

комірку можна очистити (Clear), додаток закрити (Quit), користувальницьку форму показати (Show) або сховати (Hide).

Назва методу відділяється від назви об'єкта крапкою: об'єкт.метод.

Приклади використання методів:

Range("B2:E2").Select – вибрати діапазон комірок B2:E2;

Range("C1:C5").Clear – очистити діапазон комірок C1:C5;

UserForm1.Hide – сховати форму 1;

UserForm2.Show – показати форму 2;

Application.Quit – вийти з додатка.

Властивості. Властивостями описуються характеристики об'єктів. Наприклад, розмір і колір шрифту, положення форми на екрані або стан об'єкта (доступність, видимість). Щоб змінити характеристику об'єкта, треба просто змінити значення властивості, тобто привласнити йому певні значення.

Синтаксис установлення значення властивості:

Об'єкт.Властивість = Значення_властивості,

де Об'єкт означає ім'я об'єкта, Властивість – ім'я властивості.

Ім'я об'єкта відділяється від імені властивості крапкою.

Приклади властивостей:

Range("E1").Value = 2005 - помістити в комірку E1 значення 2005;

Range("C1:C10").Text = "Основи інформаційних технологій та програмування" – помістити в діапазон комірок C1:C10 текст “Основи інформаційних технологій та програмування”;

Range("A2").Font.Size = 16 – у комірці A2 установити розмір шрифту 16.

Методи й властивості деяких об'єктів VBA

Об'єкт: Application (додаток).

Метод: Quit (закрити).

Властивість: Caption (ім'я головного вікна).

Приклади:

Application.Quit - закрити додаток;

Application.Caption = "Список" – установити як заголовок вікна додатка “Список”.

Об'єкт: Sheet (лист),
ActiveSheet (активний/обраний лист).
Сімейство: Sheets (листи).
Методи: Select (вибрати);
ShowDataForm (показати вбудовану форму).

Приклади

1 Sheets("Список").Select - вибрати лист "Список";
ActiveSheet.ShowDataForm - на активному в даний момент аркуші показати вбудовану форму.

Об'єкт: Range (діапазон).

Методи: Select (виділити);

Clear (очистити).

Властивість: Name (ім'я).

Sheets("Розрахунки").Range("B4:B10").Name = "Група" – діапазону B4:B10, розташованому на листі "Розрахунки", присвоїти ім'я "Група";

Sheets("Розрахунки").Range("B4:B10").Select – виділити діапазон B4:B10 на листі "Розрахунки".

2 Об'єкт, Сімейство: UserForm (користувальницька форма).

Методи: Show (показати);

Hide (сховати).

Властивість: Caption (текст, відображуваний у рядку заголовка).

UserForm1.Show – показати першу користувальницьку форму;

UserForm1.Hide – сховати першу користувальницьку форму;

UserForm1.Caption – "Програмування" – вивести заданий у лапках текст у рядку заголовка.

Об'єкт, Сімейство: TextBox (Поле введення).

Властивість: Text (вміст).

UserForm1.TextBox1.Text = Date – у перше поле введення у першій користувальницькій формі записати поточну дату;

UserForm1.TextBox2.Text = " " – очистити друге поле введення в першій користувальницькій формі.

Той факт, що деякі об'єкти Excel включені в інші об'єкти, має першорядну важливість і додає в об'єктну модель дуже корисну структуру.

У процесі читання різних джерел створюється враження, що не існує однозначної відповідності між батьківськими й дочірніми об'єктами. У нашому випадку, якщо об'єкт А має властивість або метод, що повертає об'єкт В, то об'єкт А буде батьківським відносно об'єкта В. Наприклад, об'єкт Range має властивість Font, що повертає об'єкт Font. Відповідно, об'єкт Range є батьківським стосовно об'єкта Font. При цьому останній є дочірнім також відносно об'єкта ChartArea. Тобто кількість предків і нащадків об'єкта необмежена.

Об'єкт Range є дочірнім відносно об'єкта Worksheet, а об'єкт Worksheet – відносно об'єкта Range. У більшості моделей багато об'єктів мають властивості, які повертають найвищий об'єкт в ієрархії моделі. В об'єктній моделі Excel практично кожен об'єкт має властивість Application, що повертає об'єкт Application, розташований на верхньому рівні ієрархії. Відповідно, практично кожен об'єкт у цьому випадку можна вважати батьківським відносно об'єкта Application.

Завдання 1

Побудувати масив – таблицю множення та знайти середнє арифметичне його елементів.

Програму наведено:

```
Private Sub CommandButton1_Click()  
Dim X  
Dim Y  
Dim M(5)  
M(1) = "A": M(2) = "B": M(3) = "C": M(4) = "D": M(5) = "E"  
S = 0: K = 0  
For i = 1 To 5 Step 1  
For j = 1 To 4 Step 1  
X = M(i) & j  
Range(X).Select  
ActiveCell.FormulaR1C1 = i * j  
S = S + ActiveCell.FormulaR1C1
```

```
K = K + 1
Next j
Next i
SR = S / K
TextBox1.Text = SR
```

```
End Sub
```

На рисунку 11.1 наведено результат розрахунків.

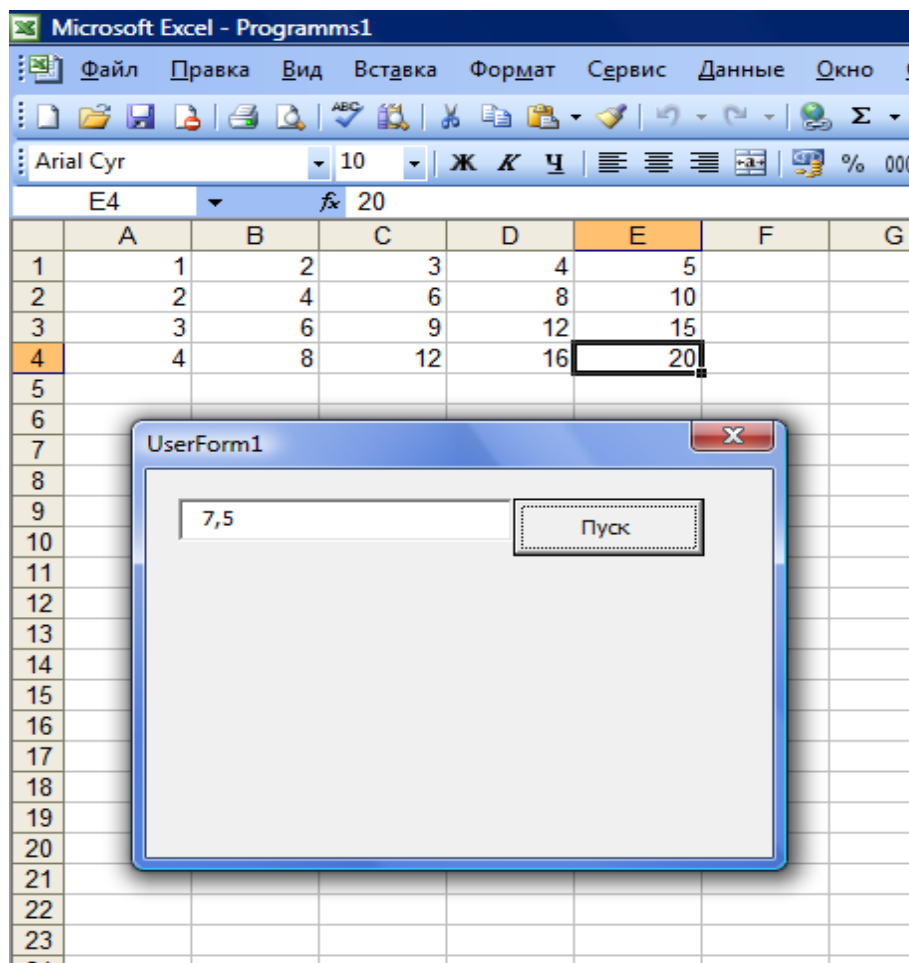


Рисунок 11.1 – Результат виконання програми

Завдання 2

Набрати на листі 1 у перших двох стовпцях таблицю вихідних даних, наприклад прогноз температури на 10 днів (A2:A11, B2:B11), перший рядок використовувати для підписування заголовків стовпців. У комірку B14 занести кількість точок у масивах (10).

На листі 1 розмістити 3 командних кнопки, одну підписати «ЧИТАННЯ», другу «ЗАПИС НА ЛИСТ2», третю «ОЧИЩЕННЯ» (рисунок 11.2).

Через кнопку «ЧИТАННЯ» відкрити редактор VBA та написати програму зчитування даних з листа 1 в змінні VBA (X, Y, N).

Через кнопку «ЗАПИС НА ЛИСТ2» відкрити редактор VBA і набрати програму запису даних на лист 2.

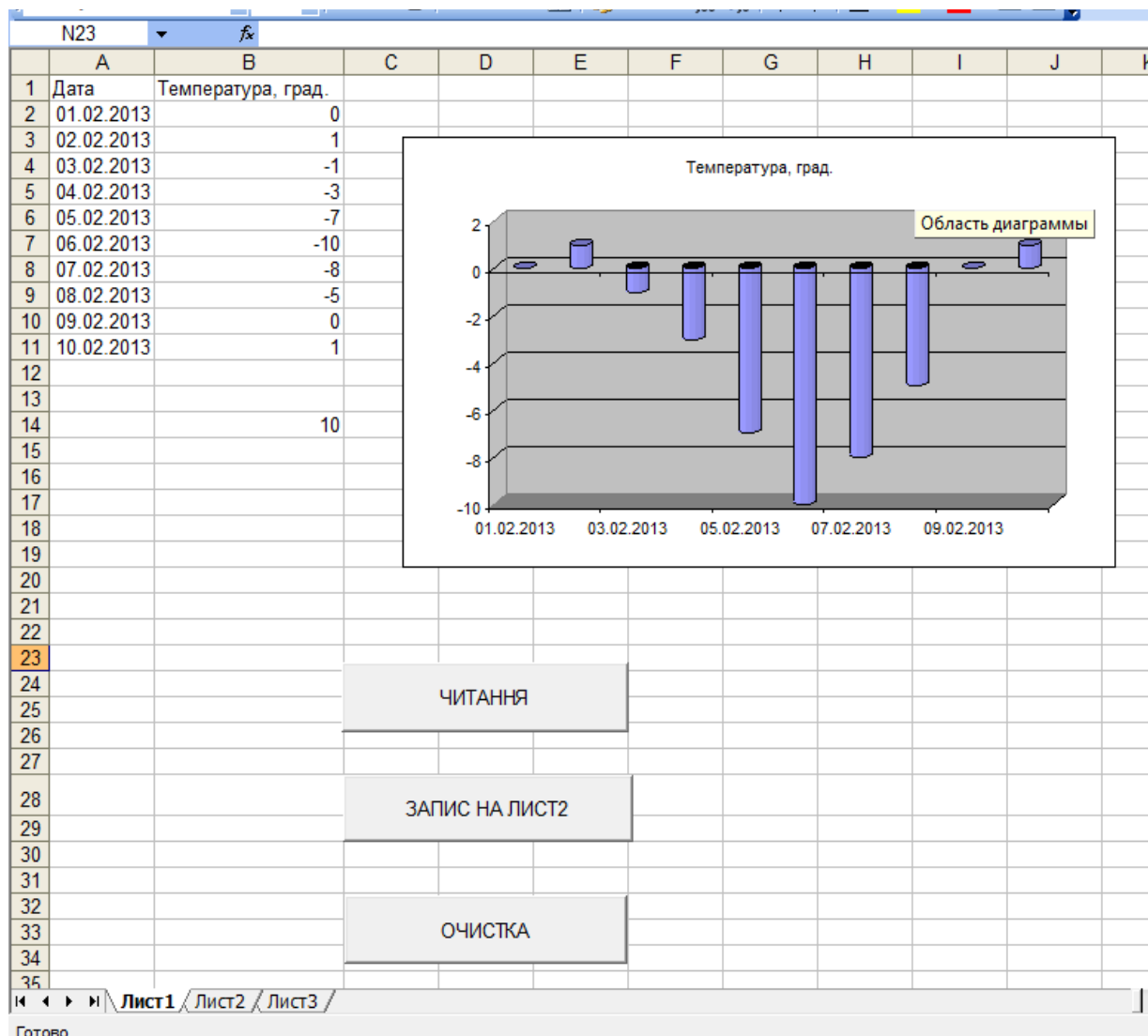


Рисунок 11.2

Через кнопку «ОЧИЩЕННЯ» набрати програму, яка очищує всі результати з листа 2.

Змінити вихідні дані на листі 1 та перевірити, що ці зміни відображаються на листі 2 після натискання відповідних кнопок.

Програму наведено:

```
Dim X(10), Y(10), N As Variant ' об'явлення змінних
Private Sub CommandButton1_Click()
' читання вмісту комірки B14 листа 1 в змінну N
N = Worksheets(1).Range("B14").Value
' цикл зчитування вмісту N комірок першого і
' другого стовпця листа 1 в масиви X і Y
For i = 1 To N
    X(i) = Worksheets(1).Cells(i + 1, 1).Value
    Y(i) = Worksheets(1).Cells(i + 1, 2).Value
Next i
End Sub
```

```
Private Sub CommandButton2_Click()
' Запис на лист 2 вмісту змінних N (у комірку D14),
' X и Y ( 3 і 4 стовпці)
Worksheets(2).Range("D14").Value = N
For i = 1 To N
Worksheets(2).Cells(i + 1, 3).Value = X(i)
Worksheets(2).Cells(i + 1, 4).Value = Y(i)
Next i
End Sub
```

```
Private Sub CommandButton3_Click()
' Очищення на листі 2 комірки D14, 3 і 4 стовпця
Worksheets(2).Range("D14").Value = " "
For i = 1 To N
Worksheets(2).Cells(i + 1, 3).Value = " "
Worksheets(2).Cells(i + 1, 4).Value = " "
Next i
End Sub
```

Зміст звіту

У звіті має бути:

- 1) мета роботи;
- 2) зроблено завдання 1 та 2;

- 3) перелічені об'єкти, що були вивчені у даній лабораторній роботі, їх властивості та методи;
- 4) перелічені основні способи посилань з одного листа на інші листи книги;
- 5) наведено аналіз роботи програм мовою Visual Basic for Applications;
- 6) висновки з лабораторної роботи.

Тестові питання до лабораторної роботи 11

Оберіть правильну відповідь (або декілька правильних відповідей):

1 Об'єктом VBA є:

- а) робоча книга;
- б) робочий лист;
- в) діапазон комірок;
- г) процесор.

2 Натискання мишею або натискання клавіші, переміщення миші або вихід із програми, ця дія називається:

- а) відгуком;
- б) переглядом;
- в) властивістю;
- г) подією.

3 Продовжите визначення: Метод – це...:

- а) основний елемент мови VBA;
- б) структура редактора VBA;
- в) характеристика об'єкта;
- г) дії, що здійснюються над об'єктами.

4 Щоб стандартно змінити характеристику об'єкта, треба:

- а) написати нову програму;
- б) застосувати метод;
- в) змінити значення властивості.

5 Скрить форму № 6:

- а) UserForm6.Show;
- б) UserForm №6.Close;
- в) UserForm6.Hide;
- г) UserForm №6.Show.

6 Вийти з додатка:

- а) Application.Hide;

- б) Application.Close;
- в) Application.Select;
- г) Application.Quit.

7 Очистити значення поля для поля введення зі списком номер три в користувальницькій формі номер два:

- а) UserForm3.ComboBox2.Text = "Очистити";
- б) UserForm2.ComboBox3.Text = " ";
- в) UserForm3.ComboBox2.Capture = "0 ";
- г) UserForm2.ComboBox3.Text = "-".

8 Установити прапорець номер п'ять у користувальницькій формі номер два :

- а) UserForm5.CheckBox2.Value = True;
- б) UserForm2.CheckBox2.Text = False;
- в) UserForm2.ComboBox5. Capture = " True ";
- г) UserForm2.CheckBox5. Value = True.

9 Скинути прапорець номер п'ять у користувальницькій формі номер два:

- а) UserForm5.CheckBox2. Value = True;
- б) UserForm2. CheckBox5. Value = False;
- в) UserForm2.ComboBox5. Capture = " True ";
- г) UserForm2.CheckBox5. Text = False.

10 Вибрати перемикач номер два в користувальницькій формі номер три:

- а) UserForm3.OptionButton2.Value = False;
- б) UserForm2. CheckBox3. Value = False;
- в) UserForm2.ComboBox5. Capture = True;
- г) UserForm3.OptionButton2.Value = True.

СПИСОК ЛІТЕРАТУРИ

- 1 Гарнаев А.Ю. Microsoft Excel 2002: Разработка приложений. – С.Пб.: Изд-во ВHV-Санкт-Петербург, 2002. – 763 с.
- 2 Гарнаев А.Ю. Самоучитель VBA. 2-е издание. – С.Пб.: Изд-во ВHV-Петербург, 2004. – 540 с.
- 3 Гарнаев А.Ю. VBA: в подлиннике, СПб.: Изд-во 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 / Укл. 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. – 1-е изд. – С.Пб.: 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.