

Лабораторна робота № 2.

Тема: *Масиви.*

Мета роботи:

Вміти застосовувати масиви при програмуванні мовою C#.

Теоретичні відомості

Оператори циклів

for (ініціалізація, умова_виходу, ітерація)
оператор;

while (умова_продовження) оператор;

do оператор; **while** (умова продовження);

Цикл **foreach** використовується для перебору об'єктів з деякої групи даних, наприклад, масиву, списку або іншого контейнера.

foreach (тип змінна **in** колекція) оператор;

Оператори передачі керування

У C# є п'ять операторів, що змінюють порядок виконання обчислення:

- оператор безумовного переходу goto;
- оператор виходу із циклу break ;
- оператор перехід до наступної ітерації циклу continue ;
- оператор повернення з функції return ;
- оператор генерації виключення throw.

Масиви

Масив — це обмежена сукупність однотипних величин. Елементи масиву мають те саме ім'я, а відрізняються за порядковим номером (*індексом*). Масив у мові C # відноситься до посилальних типів даних, тобто розташовується в динамічній області пам'яті, тому створення масиву починається з виділення пам'яті під його елементи. Елементами масиву можуть бути величини як значимих, так і посилальних типів (у тому числі масиви). Усім елементам при створенні масиву присвоюється значення за замовчуванням: нулі для значимих типів і null для посилальних.

Одномірні масиви використовуються в програмах найчастіше. Варіанти опису масиву:

тип[] ім'я;

тип[] ім'я = new тип [розмірність];

тип[] ім'я = { список_ініціалізаторів };

тип[] ім'я = new тип [] { список_ініціалізаторів };

тип[] ім'я = new тип [розмірність] { список_ініціалізаторів };

Прямокутний масив має більш одного виміру. Найчастіше в програмах використовуються двовимірні масиви. Варіанти опису двовимірного масиву:

```
тип[,] ім'я;  
тип[,] ім'я = new тип [розмірність1, розмірність2];  
тип[,] ім'я = { список_ініціалізаторів };  
тип[,] ім'я = new тип [,] { список_ініціалізаторів };  
тип[,] ім'я = new тип [розмірність1, розмірність2 ]  
{список_ініціалізаторів};
```

Східчасті (вкладені) масиви

У східчастих(**вкладених**) масивах кількість елементів у різних рядках може різнитися. У пам'яті східчастий масив зберігається інакше, чому прямокутний: у вигляді декількох внутрішніх масивів, кожен з яких має свій розмір. Крім того, виділяється окрема область пам'яті для зберігання посилань на кожен із внутрішніх масивів.

Варіанти опису двовимірного масиву:

```
тип[][] ім'я;  
тип[][] ім'я = { список_ініціалізаторів };  
тип[][] ім'я = new тип [розмірність1 ] [] {  
new тип [розмірність2_1 ] { список_ініціалізаторів},  
new тип [розмірність2_2 ] { список_ініціалізаторів},  
};
```

Таблиця 6.1. Деякі елементи класу Array

Елемент	Вид	Опис
Length	Властивість	Кількість елементів масиву (по всім розмірностям)
Binarysearch	Статичний метод	Двійковий пошук у впорядкованому масиві
Clear	Статичний метод	Присвоювання елементам масиву значень за замовчуванням
Copy	Статичний метод	Копіювання заданого діапазону елементів одного масиву в інший масив
GetValue	Метод	Одержання значення елемента масиву
IndexOf	Статичний метод	Пошук першого входження елемента в одномірний масив
Reverse	Статичний метод	Зміна порядку проходження елементів на зворотний
Sort	Статичний метод	Впорядкування елементів одномірного масиву

Методи

Метод — це функціональний елемент класу, який реалізує обчислення або інші дії, виконувані класом або екземпляром. Методи визначають поведінку класу. Метод є закінченим фрагментом коду, до якого можна звернутися по імені. Він описується один раз, а викликатися може стільки раз, скільки необхідно. Той самий метод може обробляти різні дані, передані йому в якості аргументів. Методи визначаються в класах.

Синтаксис методу:

```
[ атрибути ] [ специфікатори ] тип ім'я_методу ( [ параметри ] )  
    тіло_методу
```

Приклад 1.

Замінити всі додатні елементи відповідними від'ємними їм числами.

Варіант 1: для одновимірного масиву.

```
using System;  
namespace ConsoleApplication2  
{  
    class Class  
    {  
        static int [] Input ()  
        {  
            Console.WriteLine("Розмірність масиву");  
            int n=int.Parse(Console.ReadLine());  
            int []a=new int[n];  
            for (int i = 0; i < n; ++i)  
            {  
                Console.Write("a[{0}]= ", i);  
                a[i]=int.Parse(Console.ReadLine());  
            }  
            return a;  
        }  
  
        static void Print(int[] a)  
        {  
            for (int i = 0; i < a.Length; ++i) Console.Write("{0}  
", a[i]);  
            Console.WriteLine();  
        }  
  
        static void Change(int[] a)  
        {  
            for (int i = 0; i < a.Length; ++i)  
                if (a[i] > 0) a[i] = -a[i];  
        }  
  
        static void Main()  
        {  
            int[] myArray=Input();  
            Console.WriteLine("Вихідний масив:");  
        }  
    }  
}
```

```
        Print(myArray);
        Change(myArray);
        Console.WriteLine("Змінений масив:");
        Print(myArray);
    }
}
}
```

Варіант 2: для двовимірного масиву.

```
using System;
namespace ConsoleApplication
{
    class Class
    {
        static int [,] Input (out int n, out int m)
        {
            Console.WriteLine("Розмірність масиву ");
            Console.Write("n = ");
            n=int.Parse(Console.ReadLine());
            Console.Write("m = ");
            m=int.Parse(Console.ReadLine());
            int [,]a=new int[n, m];
            for (int i = 0; i < n; ++i)
                for (int j = 0; j < m; ++j)
                {
                    Console.Write("a[{0},{1}]= ", i, j);
                    a[i, j]=int.Parse(Console.ReadLine());
                }
            return a;
        }

        static void Print(int[,] a)
        {
            for (int i = 0; i < a.GetLength(0); ++i, Console.WriteLine() )
                for (int j = 0; j < a.GetLength(1); ++j)
                    Console.Write("{0,5} ", a[i, j]);
        }

        static void Change(int[,] a)
        {
            for (int i = 0; i < a.GetLength(0); ++i)
                for (int j = 0; j < a.GetLength(1); ++j)
                    if (a[i, j] > 0) a[i, j] = -a[i, j];
        }

        static void Main()
        {
            int n,m;
            int[,] myArray=Input(out n, out m);
            Console.WriteLine("Вихідний масив:");
            Print(myArray);
            Change(myArray);
            Console.WriteLine("Змінений масив:");
            Print(myArray);
        }
    }
}
```

Приклад 2. Підрахувати кількість максимальних елементів.

```
using System;
namespace ConsoleApplication
{
    class Class
    {
        static int [] Input ()
        {
            Console.WriteLine("Розмірність масиву ");
            int n=int.Parse(Console.ReadLine());
            int []a=new int[n];
            for (int i = 0; i < n; ++i)
            {
                Console.Write("a[{0}]= ", i);
                a[i]=int.Parse(Console.ReadLine());
            }
            return a;
        }

        static int Max(int[] a)
        {
            int max=a[0];
            for (int i = 1; i < a.Length; ++i)
                if (a[i] > max) max=a[i];
            return max;
        }

        static void Main()
        {
            int[] myArray=Input();
            int max=Max(myArray);
            int kol=0;
            for (int i=0; i<myArray.Length;++i)
                if (myArray[i]==max)++kol;
            Console.WriteLine("Кількість максимальних елементів = "+kol);
        }
    }
}
```

Приклад 3. Підрахувати середнє арифметичне непарних елементів, розташованих вище головної діагоналі.

```
using System;
namespace ConsoleApplication
{
    class Class
    {
        static int [,] Input (out int n)
        {
            Console.WriteLine("Розмірність масиву ");
            Console.Write("n = ");
            n=int.Parse(Console.ReadLine());
        }
    }
}
```

```

        int [,]a=new int[n, n];
        for (int i = 0; i < n; ++i)
            for (int j = 0; j < n; ++j)
            {
                Console.Write("a[{0},{1}]= ", i, j);
                a[i, j]=int.Parse(Console.ReadLine());
            }
        return a;
    }

    static void Print(int[,] a)
    {
        for (int i = 0; i < a.GetLength(0); ++i,Console.WriteLine() )
            for (int j = 0; j < a.GetLength(1); ++j)
                Console.Write("{0,5} ", a[i, j]);
    }

    static double Rezalt(int[,] a)
    {
        int k=0;
        double s=0;
        for (int i = 0; i < a.GetLength(0); ++i)
            for (int j = i+1; j < a.GetLength(1); ++j)
                if (a[i, j] %2!= 0) {++k; s+=a[i, j];}
        if (k!=0) return s/k;
        else return 0;
    }

    static void Main()
    {
        int n;
        int[,] myArray=Input(out n);
        Console.WriteLine("Початковий масив:");
        Print(myArray);
        double rez=Rezalt(myArray);
        Console.WriteLine("Средне арифметичне ={0:f2}", rez);
    }
}

```

Приклад 4. Дано масив розміром $n \times m$, елементи якого цілі числа. Знайти максимальний елемент у кожному рядку і записати дані в новий масив.

```

using System;
namespace ConsoleApplication
{
    class Class
    {
        static int [][] Input ()
        {
            Console.WriteLine("Розмірність масиву ");
            Console.Write("n = ");
            int n=int.Parse(Console.ReadLine());
            int [][]a=new int[n][];
            for (int i = 0; i < n; ++i)

```

```
{
    a[i]=new int [n];
    for (int j = 0; j < n; ++j)
    {
        Console.Write("a[{0},{1}]= ", i, j);
        a[i][j]=int.Parse(Console.ReadLine());
    }
}
return a;
}

static void Print1(int[] a)
{
    for (int i = 0; i < a.Length; ++i)
        Console.Write("{0,5} ", a[i]);
}

static void Print2(int[][] a)
{
    for (int i = 0; i < a.Length; ++i,Console.WriteLine()
)
        for (int j = 0; j < a[i].Length; ++j)
            Console.Write("{0,5} ", a[i][j]);
}

static int Max(int[] a)
{
    int max=a[0];
    for (int i = 1; i < a.Length; ++i)

        if (a[i] >max) {max=a[i];}
    return max;
}

static void Main()
{
    int[][] myArray=Input();
    Console.WriteLine("Вихідний масив:");
    Print2(myArray);
    int[] rez=new int [myArray.Length];
    for (int i=0;i<myArray.Length; ++i)
        rez[i]=Max(myArray[i]);
    Console.WriteLine("Новий массив:");
    Print1(rez);
}
}
```

Завдання до лабораторної роботи:

Порядок виконання роботи:

- 1) Написати C# програми, що реалізує задачі згідно з варіантом завдання. Вихідні дані вводяться із клавіатури. Завдання 1 – 4.
- 2) Реалізувати програму мовою C# відповідно до варіанта виконання.
- 3) Розробити та порівняти з виконанням програму на мовою C#.

4) Підготувати звіт у твердій копії та в електронному виді.

Варіанти завдань

Зауваження. Завдання з масивами вирішити двома способами, використовуючи одновимірний масив, а потім двовимірний. Розмірність масиву вводиться з клавіатури.

Завдання 1. Варіанти задач. Задано масив.

- 1.1. Замінити всі елементи, менші заданого числа, цим числом.
- 1.2. Замінити всі елементи, що потрапляють в інтервал, нулем.
- 1.3. Замінити всі від'ємні не кратні елементи, протилежними їм числами.
- 1.4. Всі елементи, менші заданого числа, збільшити в два рази.
- 1.5. Підрахувати середнє арифметичне елементів.
- 1.6. Підрахувати середнє арифметичне від'ємних елементів.
- 1.7. Підрахувати кількість непарних елементів.
- 1.8. Підрахувати суму елементів, що потрапляють в заданий інтервал.
- 1.9. Підрахувати суму елементів, кратних 9.
- 1.10. Підрахувати кількість елементів, що не потрапляють в заданий інтервал.
- 1.11. Підрахувати суму квадратів парних елементів.
- 1.12. Вивести на екран номери всіх елементів більших заданого числа.
- 1.13. Вивести на екран номери всіх непарних елементів.
- 1.14. Вивести на екран номери всіх елементів, які не діляться на 7.
- 1.15. Вивести на екран номери всіх елементів, які не діляться на 3 і 2.
- 1.16. Вивести на екран номери всіх елементів, що не потрапляють в заданий інтервал.
- 1.17. Визначити, чи є добуток елементів тризначним числом.
- 1.18. Визначити, чи є сума елементів двозначним числом.
- 1.19. Вивести на екран елементи з парними індексами (для двовимірного масиву - сума індексів повинна бути парною).
- 1.20. Вивести на екран додатні елементи з непарними індексами (для двовимірного масиву - перший індекс повинен бути непарним).

Зауваження. При вирішенні завдання використовувати одновимірний масив. Розмірність масиву вводиться з клавіатури.


Завдання 2. Варіанти задач. Дана послідовність з n дійсних чисел.

- 2.1. Вивести на екран номери всіх мінімальних елементів.
- 2.2. Замінити всі максимальні елементи нулями.
- 2.3. Замінити всі мінімальні елементи на протилежні.
- 2.4. Поміняти місцями максимальний елемент і перший.
- 2.5. Вивести на екран номери всіх елементів, які не збігаються з максимальним.
- 2.6. Знайти номер першого мінімального елемента.

- 2.7. Знайти номер останнього максимального елемента.
- 2.8. Підрахувати суму елементів, розташованих між максимальним і мінімальним елементами (мінімальний і максимальний елементи в масиві єдині). Якщо максимальний елемент зустрічається пізніше мінімального, то вивести повідомлення про це.
- 2.9. Знайти номер першого максимального елемента.
- 2.10. Знайти номер останнього мінімального елемента.
- 2.11. Підрахувати суму елементів, розташованих між першим і останнім максимальним мінімальними елементами. Якщо максимальний елемент зустрічається пізніше мінімального, то вивести повідомлення про це.
- 2.12. Поміняти місцями перший мінімальний і максимальний останній елементи.
- 2.13. Знайти максимум з від'ємних елементів.
- 2.14. Знайти мінімум з додатніх елементів.
- 2.15. Знайти максимум з модулів елементів.
- 2.16. Знайти кількість пар сусідніх елементів, різниця між якими дорівнює заданому числу.
- 2.17. Підрахувати кількість елементів, значення яких більше значення попереднього елемента.
- 2.18. Підрахувати кількість елементів, значення яких більше значення наступного елемента.
- 2.19. Знайти кількість пар сусідніх елементів, в яких попередній елемент кратний наступному.
- 2.20. Знайти кількість пар сусідніх елементів, в яких попередній елемент менше наступного.

Зауваження. При вирішенні завдання використовувати двовимірний масив.
Завдання 3. Варіанти задач. Дано масив розміром $n \times n$, елементи якого цілі числа.

- 3.1. Підрахувати середнє арифметичне парних елементів, розташованих нижче головної діагоналі.
- 3.2. Підрахувати суму елементів, розташованих на побічній діагоналі.
- 3.3. Підрахувати середнє арифметичне ненульових елементів, розташованих над побічній діагоналлю.
- 3.4. Підрахувати середнє арифметичне елементів, розташованих під побічній діагоналлю.
- 3.5. Поміняти місцями стовпці за правилом: перший з останнім, другий з передостаннім і т.д.
- 3.6. Поміняти місцями дві середніх рядки, якщо кількість рядків парне, і першу з середньою рядком, якщо кількість рядків непарна.

- 3.7. Поміняти місцями два середніх стовпця, якщо кількість стовпців парна, і перший з середнім стовпцем, якщо кількість стовпців непарна.
- 3.8. Якщо кількість рядків у масиві парна, то поміняти рядки місцями за правилом: перший рядок з другим, третій - з четвертим і т.д. Якщо кількість рядків у масиві непарна, то залишити масив без змін.
- 3.9. Якщо кількість стовпців у масиві парна, то поміняти стовпці місцями за правилом: перший стовпець з другим, третій - з четвертим і т.д. Якщо кількість стовпців у масиві непарна, то залишити масив без змін.
- 3.10. Обчислити A^n , n де - натуральне число.
- 3.11. Підрахувати норму матриці за формулою
$$\|A\| = \sum_i \max_j a_{i,j}.$$
- 3.12. Підрахувати норму матриці за формулою
$$\|A\| = \sum_j \max_i a_{i,j}.$$
- 3.13. Вивести елементи матриці в наступному порядку: 
- 3.14. З'ясувати, чи є матриця симетричною відносно головної діагоналі.
- 3.15. Обчислити $A * X$, де A - двовимірна матриця, X - вектор.

Завдання 4. Варіанти задач. Дано східчастий масив з n рядків, у рядках по m_j ($j=1..n$) елементів.

- 4.1. Знайти мінімальний елемент в кожному стовпці і записати дані в новий масив.
- 4.2. Для кожного рядка підрахувати кількість додатних елементів і записати дані в новий масив.
- 4.3. Для кожного стовпця підрахувати суму від'ємних елементів і записати дані в новий масив.
- 4.4. Для кожного стовпця підрахувати суму парних додатних елементів і записати дані в новий масив.
- 4.5. Для кожного рядка підрахувати кількість елементів, більших заданого числа, і записати дані в новий масив.
- 4.6. Для кожного стовпця знайти перший додатний елемент і записати дані в новий масив.
- 4.7. Для кожного рядка знайти останній парний елемент і записати дані в новий масив.
- 4.8. Для кожного стовпця знайти номер останнього непарного елемента і записати дані в новий масив.
- 4.9. Для кожного рядка знайти номер першого від'ємного елемента і записати дані в новий масив.

- 4.10. Для кожного рядка знайти суму елементів з номерами від k_1 до k_2 і записати дані в новий масив.
- 4.11. Для кожного стовпця знайти добуток елементів з номерами від k_1 до k_2 і записати дані в новий масив.
- 4.12. Для кожного рядка підрахувати суму елементів, що не потрапляють в заданий інтервал, і записати дані в новий масив.
- 4.13. Підрахувати суму елементів кожного рядка і записати дані в новий масив. Знайти максимальний елемент нового масиву.
- 4.14. Парні стовпці таблиці замінити на вектор X .
- 4.15. Непарні рядки таблиці замінити на вектор X .

Контрольні питання

1. Перерахуйте способи опису масивів.
2. Чим відрізняється зберігання в пам'яті масивів з величин типу значення та типу посилання?
3. Чи є розмірність масиву частиною опису?
4. Чи може розмірність масиву описана змінної (а не сталою)?
5. Чи можна змінити розмірність масиву після виділення пам'яті під нього?
6. Як вид масивів використовуються у C#?
7. Що відбувається, якщо кількість ініціалізаторів масиву не відповідає заявленій розмірності?
8. Що відбувається при присвоюванні масивів?
9. Опишіть два-три методи впорядкування масивів.
10. Опишіть основні методи й властивості класу System.Array.
11. Які обмеження має оператор foreach у порівнянні з оператором for?