

МАСИВИ

Масив – набір елементів однакового типу, звертатися до яких можна за одним спільним ім'ям. При оголошенні масиву використовують наступний синтаксис:

`<тип> [] <ім'я_масиву>;`

Після оголошення масиву необхідно за допомогою оператора **new** задати його розмірність.

```
int[] myArray;  
myArray = new int[10]; //масив цілих з 10 елементів  
double[] NumArr = new double[5]; // масив з 5ти дійсних елем.
```

```
int[] nums2 = new int[4] { 1, 2, 3, 5 };  
  
int[] nums3 = new int[] { 1, 2, 3, 5 };  
  
int[] nums4 = new[] { 1, 2, 3, 5 };  
  
int[] nums5 = { 1, 2, 3, 5 };
```


Клас System.Array

Всі масиви C# є нащадками класу **System.Array** у якого є ряд корисних властивостей та методів.

Length – повертає кількість елементів масиву;

Rank – повертає розмірність масиву (1 для одновимірного і т.д.).

Copy() – копіює заданий діапазон значень з одного масиву в інший;

Clone() – створює повну копію масиву разом зі значеннями;

IndexOf(), – відповідно шукають перше та останнє входження елементу в масив;

LastIndexOf()

Sort() – сортує одновимірний масив за зростанням;

BinarySearch() – швидкий пошук в сортованому масиві;

Reverse() – змінює порядок слідування елементів масиву на протилежний.

Приклад програми для виводу на екран масиву

```
using System; class Program {  
    static void Main() {  
        int[] arr = { 1, 3, 5, 8, 2 };  
        for (int i = 0; i < arr.Length; i++) {  
            Console.WriteLine("arr[{0}]={1}", i, arr[i]);  
        }  
        Console.ReadKey();  
    }  
}
```

```
foreach (int i in arr) {  
    Console.WriteLine(i);  
}
```

```
arr[0]=1  
arr[1]=3  
arr[2]=5  
arr[3]=8  
arr[4]=2
```


Клас System.Array

```
int[] numbers = new int[] { 1, 2, 3, 4, 5 };  
foreach (int i in numbers)  
{  
    Console.WriteLine(i);  
}
```

```
int[] numbers = new int[] { 1, 2, 3, 4, 5 };  
for (int i = 0; i < numbers.Length; i++)  
{  
    numbers[i] = numbers[i] * 2;  
    Console.WriteLine(numbers[i]);  
}
```

Приклад програми для сортування масиву

```
using System;  
class Program {  
    static void Main() {  
        int[] arr = { 1, 8, 5, 3, 2 }; //створимо масив  
                                         //створимо копію  
  
        int[] arrNew = (int[])arr.Clone();  
        Array.Sort(arrNew);           //відсортуємо масив  
        for (int i = 0; i < arrNew.Length; i++) {  
            //виведемо відсортовані значення  
            Console.WriteLine(«arrNew[{0}]=» + i, arrNew[i]);  
        }  
        Console.ReadKey();  
    }  
}
```

```
arrNew[0]=1  
arrNew[1]=2  
arrNew[2]=3  
arrNew[3]=5  
arrNew[4]=8
```


Клас System.Array

Зміна порядку елементів і розмір масиву

```
1 int[] numbers = { -4, -3, -2, -1,0, 1, 2, 3, 4 };
2
3 // расположим в обратном порядке
4 Array.Reverse(numbers);
5
6 // уменьшим массив до 4 элементов
7 Array.Resize(ref numbers, 4);
8
9 foreach(int number in numbers)
10 {
11     Console.WriteLine($"{number} \t");
12 }
```

4 3 2 1

Метод Copy копіює частину одного масиву в інший:

```
1 int[] numbers = { -4, -3, -2, -1,0, 1, 2, 3, 4 };
2 int[] numbers2 = new int[5];
3
4 // копируем из numbers с 2-го индекса 5 элементов
5 // и поместим их в массив numbers2, начиная с 0-го индекса
6 Array.Copy(numbers, 2, numbers2, 0, 5);
7
8 foreach(int number in numbers2)
9 {
10     Console.WriteLine($"{number} \t");
11 }
```

-2 -1 0 1 2

БАГАТОВИМІРНІ МАСИВИ

Багатовимірний масив – це масив який визначається двома або більше вимірами, а до його елементів треба звертатися за допомогою двох або більше індексів. Найпростіші багатовимірні масиви – двовимірні прямокутні масиви (або матриці). Для їх оголошення використовують наступний синтаксис:

`<тип>[,] <ім'я_масиву>;`

```
int[ , ] matrix = new int[5, 7];
```

```
matrix[0,0] = 7; //лівий верхній елемент  
matrix[3,5] = 10; //десь всередині  
matrix[4,6] = 0; //правий нижній елемент
```

```
int[] nums1 = new int[] { 0, 1, 2, 3, 4, 5 };
```

```
int[,] nums2 = { { 0, 1, 2 }, { 3, 4, 5 } };
```

```
int[,] nums1;
```

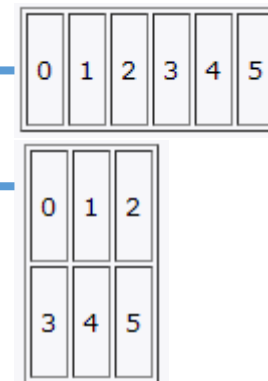
```
int[,] nums2 = new int[2, 3];
```

```
int[,] nums3 = new int[2, 3] { { 0, 1, 2 }, { 3, 4, 5 } };
```

```
int[,] nums4 = new int[,] { { 0, 1, 2 }, { 3, 4, 5 } };
```

```
int[,] nums5 = new [,] { { 0, 1, 2 }, { 3, 4, 5 } };
```

```
int[,] nums6 = { { 0, 1, 2 }, { 3, 4, 5 } };
```



БАГАТОВИМІРНІ МАСИВИ

Заповнення та виведення на екран прямокутної матриці

```
using System;
class Program {
    static void Main() {
        //створюємо матрицю
        int[,] matrix = new int[5, 7];
        //заповнюємо елементи матриці сумою індексів
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 7; j++) {
                matrix[i, j] = i + j;
            }
        }
        //виведемо отримані значення на екран
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 7; j++) {
                Console.Write("{0,3}",matrix[i, j]);
                // {0,3} - Place Holder, де 3 це
                //відстань до наступного елемента
            }
            Console.WriteLine();
        }
        Console.ReadKey();
    }
}
```

0	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10

БАГАТОВИМІРНІ МАСИВИ

Багатовимірні масиви

```
int[, ,]matrix3 = new int[5, 7, 5];  
int[, , ,]matrix4 = new int[4, 4, 3, 3];
```

Складність з перебором масиву

```
int[,] mas = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 }, { 10, 11, 12 } };  
foreach (int i in mas)  
    Console.Write($"{i} ");  
Console.WriteLine();
```

1 2 3 4 5 6 7 8 9 10 11 12

У кожного масиву є метод *GetUpperBound(dimension)*, який повертає індекс останнього елемента в певній розмірності. За допомогою виразу *mas.GetUpperBound(0) + 1* можна отримати кількість рядків таблиці, представленої двовірним масивом. Через *mas.Length / rows* можна отримати кількість елементів в кожному рядку:

```
1 int[,] mas = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 }, { 10, 11, 12 } };  
2  
3 int rows = mas.GetUpperBound(0) + 1;  
4 int columns = mas.Length / rows;  
5 // или так  
6 // int columns = mas.GetUpperBound(1) + 1;  
7  
8 for (int i = 0; i < rows; i++)  
9 {  
10     for (int j = 0; j < columns; j++)  
11     {  
12         Console.Write($"{mas[i, j]} \t");  
13     }  
14     Console.WriteLine();  
15 }
```

1 2 3
4 5 6
7 8 9
10 11 12

ЗУБЧАСТІ МАСИВИ (МАСИВ МАСИВІВ)

Зубчасті масиви (jagged array) – це масиви рядки яких можуть мати різну довжину. Причому кожен рядок зубчастого масиву є одновимірним масивом. Ці масиви оголошуються на ступним чином:

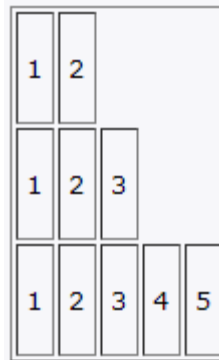
тип[][] ім'я_масиву;

```
int[ ][ ] arrJ = new int[3][ ];
```

```
arrJ[0] = new int[3];
```

```
arrJ[1] = new int[4];
```

```
arrJ[2] = new int[2];
```



```
arrJ[0][0] = 2;
```

```
arrJ[0][1] = 4;
```

```
//...
```

```
arrJ[2][1] = 17;
```

```
int[ ][ ] nums = new int[3][ ];
```

```
nums[0] = new int[2] { 1, 2 }; // выделяем память для первого подмассива
```

```
nums[1] = new int[3] { 1, 2, 3 }; // выделяем память для второго подмассива
```

```
nums[2] = new int[5] { 1, 2, 3, 4, 5 }; // выделяем память для третьего подмассива
```


МАСИВ ПАРАМЕТРІВ І КЛЮЧОВЕ СЛОВО **PARAMS**

Використовуючи ключове слово **params**, можемо передавати невизначену кількість параметрів. Сам параметр з ключовим словом **params** при визначенні методу повинен представляти одновимірний масив того типу, дані якого збираємося використовувати. При виклику методу на місце параметра з модифікатором **params** можемо передати як окремі значення, так і масив значень, або взагалі не передавати параметри.

```
1 static void Addition(params int[] integers)
2 {
3     int result = 0;
4     for (int i = 0; i < integers.Length; i++)
5     {
6         result += integers[i];
7     }
8     Console.WriteLine(result);
9 }
10
11 static void Main(string[] args)
12 {
13     Addition(1, 2, 3, 4, 5);
14
15     int[] array = new int[] { 1, 2, 3, 4 };
16     Addition(array);
17
18     Addition();
19     Console.ReadLine();
20 }
```

Якщо потрібно передати якісь інші параметри, то вони мають бути вказані до параметра з ключовим словом **params**:

```
//Так працює
static void Addition( int x, string mes, params int[] integers)
{ }
```

```
Addition(2, "hello", 1, 3, 4);
```

```
//Так НЕ працює
static void Addition(params int[] integers, int x, string mes)
{ }
```


МАСИВ ПАРАМЕТРІВ І КЛЮЧОВЕ СЛОВО **PARAMS**

```
1 // передача параметра с params
2 static void Addition(params int[] integers)
3 {
4     int result = 0;
5     for (int i = 0; i < integers.Length; i++)
6     {
7         result += integers[i];
8     }
9     Console.WriteLine(result);
10 }
11 // передача массива
12 static void AdditionMas(int[] integers, int k)
13 {
14     int result = 0;
15     for (int i = 0; i < integers.Length; i++)
16     {
17         result += (integers[i]*k);
18     }
19     Console.WriteLine(result);
20 }
```