

Масиви. Створення консольних проектів у C#



Масиви

Одновимірні масиви

Досі в задачах ми розглядали змінні, в кожному з яких можна записати лише одне значення в даний момент часу. Існує багато задач, в яких виникає необхідність збереження в пам'яті послідовності значень однієї і тієї ж величини

Наприклад, розглянемо задачу про збереження в пам'яті та обробки результатів метеорологічних вимірювань температури повітря протягом тижня.

Змінна t (температура) повинна прийняти 7 значень за кожен з днів вимірювання. Зрозуміло, можна було б описати 7 змінних: t_1, t_2, \dots, t_7 . Проте цей метод ускладнить процес обробки даних.

Масивом назвемо послідовність однотипних даних, які мають одне ім'я, а розрізняються за індексом.

Масиви

Одновимірні

Двовимірні

Багатовимірні

Масиви

Масивом називають упорядковану сукупність елементів одного типу.

Кожен елемент масиву має індекси, що визначають порядок елементів.

Індекси задаються цілочисловим типом.

Число індексів характеризує розмірність масиву.

У мові C # кожний індекс змінюється в діапазоні від 0 до деякого кінцевого значення.

Масиви в мові C # є справжніми динамічними масивами.

Ще раз про масиви



Масив - це великий простір чогось однорідного за типом. (Зі словника іноземних слів, 1954 р.)

Масив у програмуванні - це тип структури даних, що має складені значення. (З Оксфордського словника англійської мови, 1995 р.)

Масив - це впорядкований скінченний набір елементів (даних) одного типу. Зазвичай працюють з масивами, які містять числа.

Масивом називається скінченна послідовність змінних одного типу, які мають однакове ім'я та різняться порядковим номером.

Індексом називається порядковий номер елемента масиву.

Отже, введено новий тип — масив. Усі типи, які досі були вам відомі, називаються простими. Масив є прикладом структурованого типу, тобто він, у свою чергу, складається з елементів іншого типу.

Ще про масиви



- Масив являє собою структуру даних, яка дозволяє одній змінній зберігати кілька значень.
- При оголошенні масиву ви повинні вказати тип значень, що зберігаються в масиві, а також кількість значень (званих **елементами масиву**).
- Всі елементи всередині масиву повинні бути одного і того ж типу, наприклад, `int`, `float` або `char`.
- Для збереження значення всередині масиву слід вказати **номер елемента масиву**, в якому ви хочете зберегти своє значення.
- Щоб звернутися до значення, що зберігається всередині масиву, програми вказують **ім'я масиву і номер елемента**.
- При оголошенні масиву програми можуть використовувати оператор присвоювання для ініціалізації елементів масиву.
- Програми можуть передавати змінні-масиви в функції точно так само, як вони передають будь-який інший параметр.

Типи масивів



Масиви бувають **одновимірними** (у вигляді послідовності чисел), **двовимірними** (у вигляді таблиць чисел розміром $m \times n$) і **багатовимірними** (3-,4-вимірні і т.д. 3-вимірні - це об'ємний простір з комірками, а 4-вимірні і більше - це фантастично-абстрактні поняття).

Масив називається одновимірним, якщо для задання місцеположення елемента в масиві необхідно вказати значення лише одного індексу.

Масив називається двовимірним, якщо для задання місцеположення елемента в масиві необхідно вказати значення двох індексів.

Запам'ятайте, що у двовимірних масивах перший індекс завжди вказує на номер рядка, а другий - на номер стовпчика в цьому рядку!

Типові задачі



- 1) Введення масиву з клавіатури
- 2) Друк масиву на екран
- 3) Перебір всіх елементів
- 4) Пошук мінімального/максимального елеманту
- 5) Сортування за зростанням/спаданням

Оголошення масиву



Перед використанням масиву, як і будь-якої змінної його треба створити. Нехай нам потрібно створити масив `t` з кількістю 7 елементів, і кожен з елементів дійсного типу (`double`)

```
double [ ] t = new double [7];
```

Такий масив, який представляє послідовність із заданої кількості даних певного типу назвемо **одновимірним**. Для доступу до певного елемента масиву після імені масиву в квадратних дужках вказуватимемо числовий індекс. Найпершим елементом масиву в нашому випадку буде елемент з індексом 0 (нуль), тобто `t[0]`. Враховуючи, що елементів масиву 7, то останнім елементом буде елемент `t[6]`. При спробі звернутися до елемента з номером поза діапазоном `[0.6]` виникне помилка, яка може привести до аварійного переривання програми.

```
t [0] =15; //запис до 1-го елементу масиву t значення 15
```

```
t[2]=10; // запис до 3-го елементу масиву t значення 10
```

```
Console.WriteLine (t[1]); // виведення на екран значення 2-го елементу масиву t
```

Значення до масиву можна записувати як операцією присвоювання, так і задавати одночасно з його створенням.

```
double [ ] t = new double [7] {11,13 5,15,10, 9 5, 13 1,15 4};
```


Оголошення масиву чисел

```
double [] x = {5.5, 6.6, 7.7};
```

Слідуючи синтаксису, елементи константного масиву слід укласти в фігурні дужки.

Якщо масив ініціалізується константним масивом, то в динамічній пам'яті створюється константний масив із заданими значеннями, з яким і зв'язується посилання.

```
int [] d = new int [5];
```

Тут оголошено динамічний цілочисельний масив, в якому будуть зберігатися 5 цілих чисел. Масив створюється в динамічній пам'яті, його елементи отримують початкові нульові значення, і посилання зв'язується з цим масивом.

Оголошення масивів



```
// Оголошує три одновимірних масиви A, B, C
int [] A = new int [5], B = new int [5], C = new int [5];
// Заповнюється даними з клавіатури масив A
for (int i = 0; i < 5; i++) A [i] = int.Parse (Console.ReadLine ());
// Обчислюються елементи масиву C
for (int i = 0; i < 5; i++) C [i] = A [i] + B [i];
// Оголошення масиву з явною ініціалізацією      int [] x = {5.5, 6.6, 7.7};
// Оголошення масивів з відкладеним ініціалізацією
int [] u, v;
u = new int [3];
for (int i = 0; i < 3; i++) u [i] = i + 1;
// V = {1,2,3}; // присвоювання константного масиву неприпустимо!
v = new int [4];
v = u; // допустиме присвоювання - масиви одного класу
int [,] w = new int [3,5];
// V = w; // неприпустиме присвоювання: об'єкти різних класів
```

Оголошення динамічного масиву A1



```
Console.WriteLine ("Введіть число елементів масиву A1");  
int size = int.Parse (Console.ReadLine ());  
int [] A1 = new int [size];
```

Звертання до елементів масиву



Звертатися до елементів масиву можна таким чином:

a [0], b [10], c [i],

тобто спочатку вказується ім'я змінної масиву, а потім у квадратних дужках йде вказівка номера елемента.

Багатовимірні масиви

Щоб розібратися з багатовимірними масивами уявимо шахову дошку. Кожна клітинка задається двома індексами, буквою (від a до h) та цілим числом (від 1 до 8).

Шахівниця — це приклад двовимірного масиву. Процес створення двовимірного масиву схожий з одновимірним

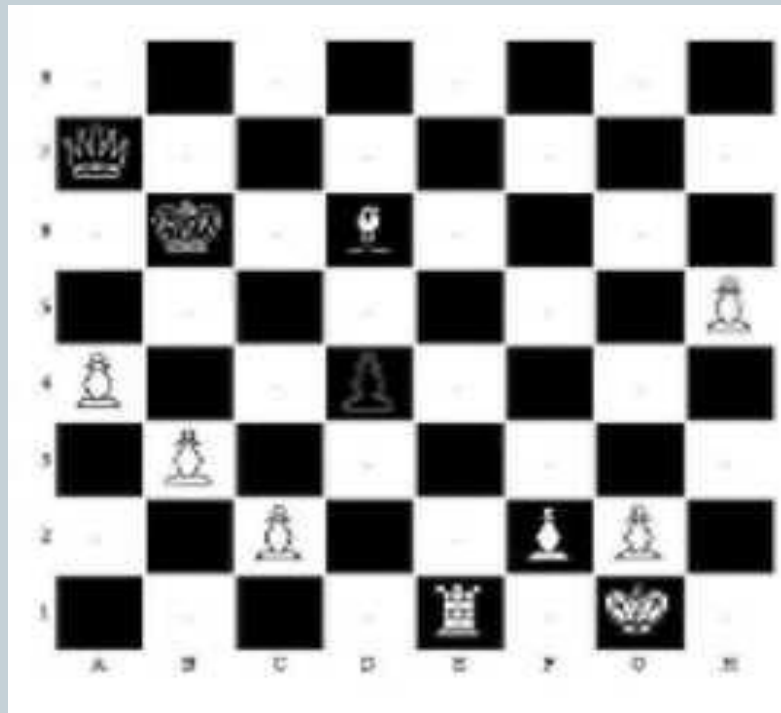
```
int [,] Array2 = new int[2, 3];
```

або

```
int [,] Array2 = {{1,2,3}, {4,5,6}};
```

Для запису значення до елементу масиву
`Array2[1,2]=43;`

Різновидом багатовимірних масивів є зубчасті масиви. У ньому кількість елементів у кожному рядку задається окремо. Прикладом задачі з використанням зубчастих масивів t є масив місяців та днів у році.



Фрагменти програм, що ілюструють різні способи створення масивів.



```
class TestArrays
{
    static void Main()
    {
        // Оголошення одновимірного масиву
        int[] array1 = new int[5];

        // Оголошення з одночасним заданням елементів одновимірного масиву
        int[] array2 = new int[] { 1, 3, 5, 7, 9 };

        // Альтернативний спосіб оголошення масиву
        int[] array3 = { 1, 2, 3, 4, 5, 6 };

        // Оголошення двовимірного масиву
        int[,] multiDimensionalArray1 = new int[2, 3];

        // Оголошення з одночасним заданням елементів двовимірного масиву
        int[,] multiDimensionalArray2 = { { 1, 2, 3 }, { 4, 5, 6 } };

        // Оголошення зубчастого масиву
        int[][] jaggedArray = new int[6][];

        // Встановлення значення елементам першого рядка зубчастого масиву
        jaggedArray[0] = new int[4] { 1, 2, 3, 4 };
    }
}
```

Первинне закріплення



1. Вкажіть рядок в якому допущено помилку при описі масиву:

a) `int [] array2 = new int [] {1,3,5,7,9};`

б) `int [] array1 = new int [5];`

в) `int [] multiDimensionalArray1 = new int [2,3];`

г) `int [] array1 = new int [5];`

2. Як називають номер елемента одновимірного масиву?

3. Як описуються та як ініціюються одновимірні масиви? Чи можна це зробити однією командою?

4. Що буде виведено на консоль у результаті виконання наступного коду?

```
int [] array2 = new int [] { 1, 3, 5, 7, 9 };  
Console.WriteLine(array2[2]+array2[3]);
```

Первинне закріплення



5. Що буде виведено на консоль у результаті виконання наступного коду?

```
int [,] array3 = {{1, 3, 5}, {0, 4, 5}, {1, 4, 3}, {0, 0, 4}};  
Console.WriteLine(array3[2, 0]+array3[1, 2]);
```

6. Чим особливим виділяються зубчасті масиви?

7. В чому відмінність описів:

```
int[] array1 = new int[5];  
int[] array1;  
int[] array1 = new int[] { 1, 3, 5, 7, 9 };
```

8. В якому з рядків описано одновимірний масив, а в якому – двовимірний?

```
int [,] array3 = {{1, 3, 5}, {0, 4, 5}, {1, 4, 3}, {0, 0, 4}};  
int [] array2 = new int [] { 1, 3, 5, 7, 9 };
```

9. Написати програму для заповнення масиву цілими числами: 6, 8, 12, 86, 0, -5, 8, 9

10. Що буде виведено на консоль у результаті виконання наступного коду?

```
int [] array2 = new int [] { -1, 3, 5, 17, 9 };  
Console.WriteLine(array2[1]+array2[3]);
```


Клас Array



Не можна зрозуміти багато деталей роботи з масивами в C#, якщо не знати пристрій класу Array з бібліотеки FCL, нащадками якого за визначенням є всі класи-масиви. Розглянемо наступні оголошення:

```
int [] ar1 = new int [5];  
double [] ar2 = {5.5, 6.6, 7.7};  
int [,] ar3 = new Int32 [3,4];
```

До якого типу, або до яких класів належать об'єкти ar1, ar2 і ar3? Всі вони належать до різних класів. Змінна ar1 належить до класу int [] - одновимірному масиву значень типу int, ar2 - double [], одновимірному масиву значень типу double, ar3 - двовимірному масиву значень типу int.

Властивості і методи класу System.Array



| Елемент | Вид | Опис |
|--------------|----------------------|---|
| Length | властивість | Повертає кількість елементів даного масиву (по всіх розмірності) |
| BinarySearch | статичний метод | Даний метод виконує двійковий пошук по відсортованому масиву |
| Clear | статичний метод | Відбувається обнуління значень масиву, тобто кожному елементу присвоюється значення за замовчуванням. |
| Copy | статичний метод | Виконує копіювання зазначеного шматка масиву з одного в інший |
| CopyTo | екземплярність метод | Копіювання всіх елементів даного масиву в інший |
| GetValue | екземплярність метод | Повертає значення даного елемента |
| IndexOf | статичний метод | Пошук першого елемента в масиві рівного даному значенню |
| LastIndexOf | статичний метод | Пошук останнього елемента в масиві рівного даному значенню |
| Reverse | статичний метод | Відбувається зміна порядку проходження елементів на зворотний |
| SetValue | екземплярність метод | Відбувається присвоєння значення вказаною елементу |
| Sort | статичний метод | Метод, який використовується для сортування елементів масиву в csharp (с #) |

Багатовимірні масиви в мові програмування

C#



Багатовимірні масиви мають більше одного виміру. Найбільш часто використовуваними представниками даних масивів є двовимірні, які так само називають матриці. Двовимірні масиви (матриці) являють собою таблиці. Нумерація рядків і стовпців, так само як і в одномірних починається з нуля.

Приклади оголошення двовимірного масиву (матриці):

```
тип [,] ім'я_масиву;
```

```
тип [,] ім'я_масиву = new тип [розмір1, розмір2];
```

```
тип [,] ім'я_масиву = {{елементи 1-го рядка}, ..., {елементи n-го рядка}};
```

```
тип [,] ім'я_масиву = new тип [,] {{елементи 1-го рядка}, ..., {елементи n-го рядка}};
```

Приклади оголошення двовимірних масивів



Конкретні приклади оголошення матриць
(двовимірних масивів) в C#:

```
int [,] a;
```

```
int [,] a = new int [3, 4];
```

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

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

Ступінчасті масиви в мові програмування C#



Ступінчасті масиви відрізняються від багатовимірних тим, що кількість елементів в різних рядках масиву може відрізнятися. У пам'яті ПК даний вид масиву зберігатися у вигляді масиву масивів.

Оголошення ступеневого масиву:

тип [] [] імя_масиву;

наприклад: int [] [] a;

Оголошено одновимірний масив посилань на цілочисельні одновимірні масиви. При такому описі потрібно не тільки виділяти пам'ять під одновимірний масив посилань, але і під кожен з цілочисельних одновимірних масивів. Такий розподіл пам'яті дозволяє визначати довільну довжину кожного рядка масиву (звідси і пішла назва масиву - ступінчастий).

Наприклад:

```
int [ ] [ ] a = new int [3] [ ]; // Створюємо три рядки
```

```
a [0] = new int [2]; // 0-ий рядок посилається на 2-х елементів одновимірний масив
```

```
a [1] = new int [3]; // 1-ий рядок посилається на 3-х елементів одновимірний масив
```

```
a [2] = new int [10]; // 2-ий рядок посилається на 10-х елементів одновимірний масив
```

Висновок



Для оголошення масиву в С# використовується наступний синтаксис:

- 1) *int [] mas = {1,2,5,6,8,4,9,10};*
- 2) *int [] mas2 = new int [10];*

Оголошувати масив в С# можна явно, а можна просто виділяти пам'ять. Аналогічним чином оголошуються інші змінні, наприклад double.

- 1) *double [] mas = {1, 2, 5, 6, 8, 4, 9, 10};*
- 2) *double [] mas2 = new double [10];*

Тривимірний масив



У наведеному прикладі показана робота з тривимірним масивом. Масив створюється з використанням циклів типу `for`, а при знаходженні суми його елементів, мінімального і максимального значення використовується цикл `foreach`:

```
int [, ,] arr3d = new int[10, 10, 10];  
for(int i = 0; i < 10; i++)  
    for(int j = 0; j < 10; j++)  
        for(int k = 0; k < 10; k++)  
            arr3d[i, j, k] = int.Parse(Console.ReadLine());  
long sum = 0;  
int min = arr3d[0, 0, 0], max = arr3d[0, 0, 0];  
foreach(int item in arr3d)  
{  
    sum += item;  
    if(item > max) max = item;  
    else if(item < min) min = item;  
}  
Console.WriteLine("sum = {0}, min = {1}, max = {2}", sum, min, max);
```

Приклад роботи з масивом



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace ConsoleApplication2
{
    class Program
    {
        static void Main (string [] args)
        {
            int [] mas = {1,2,5,6,8,4,9,10};
            int [] mas2 = new int [10];
            double [] mas3 = {1.5, 2, 5, 6.5, 8, 4, 9, 10};
            double [] mas4 = new double [10];
            Console.WriteLine ("Виведення 1-го елемента mas:" + mas [0]);
            Console.WriteLine ("Виведення 1-го елемента mas2:" + mas2 [0]);
            Console.WriteLine ("Виведення 1-го елемента mas3:" + mas3 [0]);
            Console.WriteLine ("Виведення 1-го елемента mas4:" + mas4 [0]);
            Console.ReadKey ();
        }
    }
}
```

Як бачимо, що якщо ми не вказуємо значення масиву, але виділяємо пам'ять, то за умовчанням в C# записується нуль. В програмі ми виводимо масиви с sharp за допомогою функції виводу Console.WriteLine.

Результат роботи програми:

```
Виведення 1-го елемента mas:1
Виведення 1-го елемента mas2:0
Виведення 1-го елемента mas3:1,5
Виведення 1-го елемента mas4:0
```