

Платформа .NET и язык программирования C#

Массивы

Массивы

Массив — упорядоченная совокупность однотипных данных.

Массив обязательно имеет имя.

Каждый элемент массива имеет уникальный индекс. Нумерация элементов массива начинается с нуля.

Объявление массива:

тип[] имяМассива;

Например:

```
int[] array; //объявили массив целых чисел
```

Инициализация массива:

```
имяМассива = new тип[числоЭлементовМассива];
```

Например,

```
array = new int[3]; // инициализировали массив из трех целых  
// чисел
```

Присвоение значения элементу массива

Присвоить значение элементу массива или изменить его значение можно, обратившись к элементу по его индексу.

`имяМассива[индексЭлемента] = значениеЭлемента;`

Например:

```
array[0] = 18;
```

```
array[1] = -2;
```

```
array[2] = 0;
```

Массивы (продолжение)

Короткая форма записи массива

{ элемент0, элемент1, ..., элементN }

Например, при инициализации можно записать

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

Можно даже задавать тип элементов неявно — компилятор вычислит тип из последующей инициализации:

```
var array = new[] {"abc", "ABC", ""} //массив строк из 3-х элементов
```

Пробежка по элементам массива с указанием индекса массива осуществляется с помощью цикла for. Например:

```
var array = new int[25]; //объявление и инициализация массива
for (int i = 0; i < array.Length; i++)
    array[i] = i * i; //задаем значения элементов массива
```

Цикл foreach

Если для вычислений значение индекса элемента не требуется и нам не нужно изменять элементы массива, можно использовать цикл `foreach`.

Цикл `foreach`

```
foreach (переменная in имяМассива)
{
    ... //тело цикла
}
```

Например,

```
var bookTitles = new string[] {"Война и мир", "Муму", "", "Hamlet"};
foreach( var title in bookTitles)
    Console.WriteLine(title);
```

Переменной цикла `foreach` нельзя присвоить новое значение. Изменить значение элемента массива можно только обратившись к нему по его индексу.

Тип-значение и тип-ссылка

Тип-значение (value type)	Тип-ссылка (reference type)
Объект хранится там, где он объявлен.	Объект хранится в куче. В месте, где объект объявлен, стоит ссылка на него.
Не может иметь значение null.	Может иметь значение null.
Заранее известно, сколько памяти он займет.	Заранее может быть неизвестно, сколько памяти он займет.
Объект удаляется вместе с контентом, в котором он хранится.	Объект удаляется сборщиком мусора после того, как на объект не будет ни одной актуальной ссылки.
Равенство объектов — это равенство значений	Равенство объектов — это совпадение ссылок (если для типа операция сравнения не переопределена, как, например, в типе string).

Класс Array

Каждый создаваемый массив получает большую часть функциональности от класса `System.Array`.

Полезные методы класса `Array`:

`Clone` — создает новый массив — поверхностную копию массива

`Copy` — поверхностно копирует массив в другой массив подходящего типа и размера

`Sort` — сортирует массив по умолчанию или указанному критерию

`BinarySearch` — осуществляет поиск элемента в **отсортированном** массиве

Многомерные массивы

```
тип[ , (... ,)] имяМассива;  
имя_массива = new тип[длина1, ..., длинаN];
```

Например,
`int[,] matrix = new int[2,3];` // задает и инициализирует
двумерный // массив размеров 2x3

Полезно знать методы и свойства:

`GetLength(i)` — число элементов по i-му измерению

`Length` — общее число элементов

`Rank` — число измерений

Визуализация двумерного массива: таблицы (матрицы)

<code>matrix[0, 0]</code>	<code>matrix[0, 1]</code>	<code>matrix[0, 2]</code>
<code>matrix[1, 0]</code>	<code>matrix[1, 1]</code>	<code>matrix[1, 2]</code>

массив `matrix [2,3]`

- Двумерный массив представляет собой матрицу или таблицу данных
- Он индексируется переменными `row` и `column` как `matrix [row, column]`
(индексы интерпретируются как номер строки и номер столбца соответственно)
- Первый индекс массива изменяется по вертикали
- Второй индекс массива изменяется по горизонтали

Визуализация двумерного массива: изображения

<code>image[0, 0]</code>	<code>image[1, 0]</code>	<code>image[2, 0]</code>
<code>image[0, 1]</code>	<code>image[1, 1]</code>	<code>image[2, 1]</code>

массив `image [3,2]`

- Двумерный массив представляет собой изображение, состоящее из точек (пикселей).
- Он индексируется переменными `x` и `y` как `image[x, y]` (индексы интерпретируются как целочисленные координаты точки).
- Первая координата изменяется по горизонтали
- Вторая координата изменяется по вертикали

Массивы массивов

Объявление массива массивов:

тип[][] имяМассива;

Например:

```
int[][] array; //объявили массив целочисленных массивов
```

Инициализация массива массивов:

имяМассива = new тип[числоМассивов][];

Например,

```
array = new int[3] []; //инициализировали массив из трех  
целочисленных массивов
```

Инициализация элемента массива массивов:

имяМассива[индекс] = new тип[числоЭлементов];

Обращение к элементу массива массивов:

имяМассива[индексВнешн][индексВнутр]

```
int[][] array = new int[6][];  
array[0] = new int[3];  
array[1] = new int[2];  
array[2] = new int[4];  
array[3] = new int[0];  
array[4] = new int[3];  
array[5] = new int[1];
```

array[0][0]

array[0][1]

array[0][2]

array[1][0]

array[1][1]

array[2][0]

array[2][1]

array[2][2]

array[2][3]

array[3] – пустой массив

array[4][0]

array[4][1]

array[4][2]

array[5][0]