



Web-разработка на C# и платформе Microsoft .NET

Массивы и методы

План занятия

- Одномерные, многомерные и рваные массивы
- Цикл `foreach`
- Класс `Array`
- Методы
- Виды аргументов
- Критерии создания методов



Одномерные массивы

- Объявление массива

`тип_элементов[] имя;`

- Создание массива

`имя = new тип_элементов
[число_элементов];`

- Доступ к элементам

`имя[номер_элемента]`

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

```
int[] arr;  
arr = new int[8];  
arr[0] = 5;  
arr[1] = 7;  
arr[2] = arr[1] + 1;
```

Одномерные массивы: сокращённая инициализация

- Стандартная инициализация

```
int[] arr = new int[2];  
arr[0] = 0;  
arr[1] = 1;
```

- Задание значений

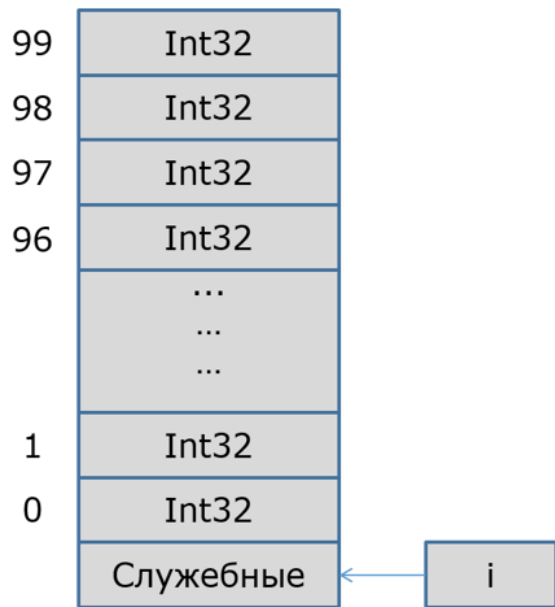
```
int[] arr = new int[2] {0, 1};  
int[] arr = new int[] {0, 1};
```

- Автоматическое определение числа элементов

```
int[] arr = {0, 1};
```

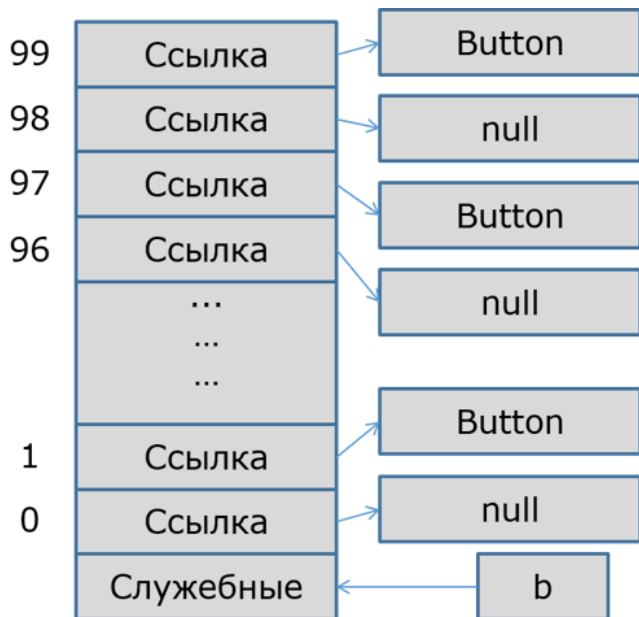
ТИПЫ-ЗНАЧЕНИЯ

```
int[] i = new int[100];
```



ССЫЛОЧНЫЕ ТИПЫ

```
Button[] b = new Button[100];
```



Ввод массива с клавиатуры

```
class Program
```

```
{  
    static void Main(string[] args)  
    {  
        int n;  
        Console.Write("Введите число элементов массива: ");  
        n = int.Parse(Console.ReadLine());  
        int[] arr = new int[n];  
        for (int i = 0; i < arr.Length; i++)  
        {  
            Console.Write("Введите arr[{0}] ", i);  
            arr[i] = int.Parse(Console.ReadLine());  
        }  
        Console.ReadLine();  
    }  
}
```

```
Введите число элементов массива: 5  
Введите arr[0] 1  
Введите arr[1] 2  
Введите arr[2] 3  
Введите arr[3] 4  
Введите arr[4] 5
```

Вывод массива на экран

```
static void Main(string[] args)
{
    Ввод массива

    Console.WriteLine("Элементы массива:");
    for (int i = 0; i < n; i++)
    {
        Console.WriteLine("arr[{0}] = {1}",
            i, arr[i]);
    }
    Console.ReadLine();
}
```

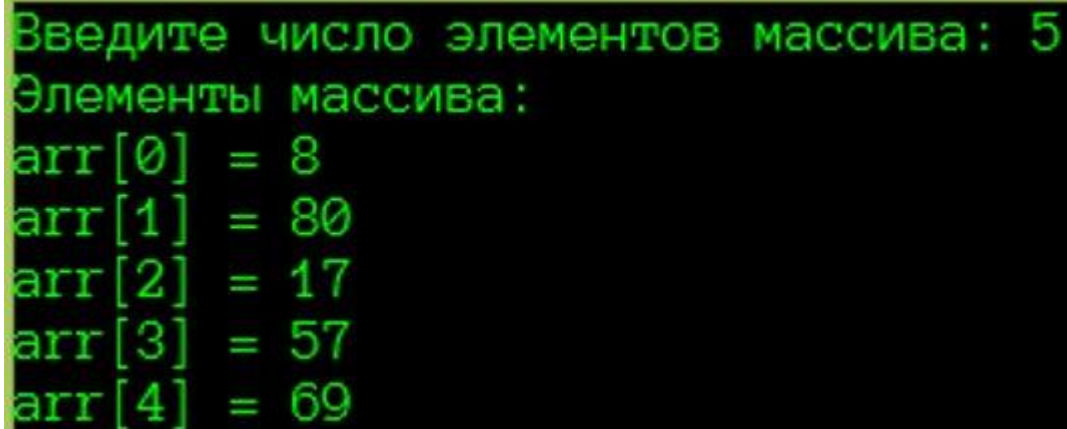
```
Введите число элементов массива: 5
Введите arr[0] 1
Введите arr[1] 2
Введите arr[2] 3
Введите arr[3] 4
Введите arr[4] 5
Элементы массива:
arr[0] = 1
arr[1] = 2
arr[2] = 3
arr[3] = 4
arr[4] = 5
```

Заполнение массива случайными числами

```
static void Main(string[] args)
{
    int n;
    Console.WriteLine("Введите число элементов
массива: ");
    n = int.Parse(Console.ReadLine());
    int[] arr = new int[n];

    Random r = new Random();
    for (int i = 0; i < arr.Length; i++)
    {
        arr[i] = r.Next(100);
    }

    Вывод массива на экран
}
```



```
Введите число элементов массива: 5
Элементы массива:
arr[0] = 8
arr[1] = 80
arr[2] = 17
arr[3] = 57
arr[4] = 69
```


Цикл foreach

- Синтаксис

```
foreach (тип_элемента имя in набор)
{
    тело_цикла
}
```

- Пример

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

Многомерные массивы

- Объявление массива
`тип_элементов[,] имя;`
- Создание массива
`имя = new тип_элементов [N, K];`
- Доступ к элементам
`имя[n, k]`

	0	1	2	3	4	5
0	5	7	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	8	0	0	0	0

```
int[,] arr;  
arr = new int[4,6];  
arr[0,0] = 5;  
arr[0,1] = 7;  
arr[3,1] = arr[0,1] + 1;
```

Рваные (jagged) массивы

- Объявление массива
`тип_элементов[][] имя;`
- Создание массива ссылок
`имя = new тип_элементов [N][];`
- Создание массивов
`имя[i] = new тип_элементов [K];`
- Доступ к элементам
`имя[n][k]`

0	0	0
---	---	---

0	0	0	0	0
---	---	---	---	---

0

0	0	0	0
---	---	---	---

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

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

Свойства и методы класса Array

- Свойства

Length

LongLength

Rank

- Методы

GetLength(размерность)

GetLowerBound(размерность)

GetUpperBound(размерность)

Статические методы класса Array

- CopyTo
- IndexOf
- LastIndexOf
- Reverse
- Sort
- Exists
- Find
- CopyTo

```
int[] arr = { 1, -2, 3, 4, -5, 2, 7, 1, 9 };
```

```
int res1 = Array.IndexOf(arr, 1);
```

```
int res2 = Array.LastIndexOf(arr, 1);
```

```
Array.Sort(arr);
```

Объявление методов

[атрибуты]

[спецификаторы] тип_результата имя_метода(список_аргументов)

{

тело_метода

[return результат]

}

Пример

```
static bool IsEven(int number)
{
    return number % 2 == 0;
}

static void Main(string[] args)
{
    Console.WriteLine("Введите целое число");
    int num = int.Parse(Console.ReadLine());
    if (IsEven(num))
    {
        Console.WriteLine("Введённое число является чётным");
    }
    else
    {
        Console.WriteLine("Введённое число является нечётным");
    }
    Console.Read();
}
```

Пример

```
static void WriteArray(int[] arr)
{
    foreach (int elem in arr)
    {
        Console.Write("\t{0}", elem);
    }
    Console.WriteLine();
}

static void Main(string[] args)
{
    int[] arr = { 5, 22, 2, 4, 6 };
    Console.WriteLine("Исходный массив");
    WriteArray(arr);
    Console.WriteLine("Сортированный массив");
    Array.Sort(arr);
    WriteArray(arr);
    Console.Read();
}
```


Именованные и необязательные аргументы

- Именованные аргументы позволяют указать значение для определённого аргумента, в зависимости от его имени, а не позиции в списке аргументов.
- Значения для необязательных аргументов можно опускать – им будет присвоено значение по умолчанию.

Пример

```
public static int Sum(int op1, int op2, int op3 = 0, int op4 = 0)
{
    return op1 + op2 + op3 + op4;
}
```

```
static void Main(string[] args)
{
    Console.WriteLine(Sum(1, 2, 3, 4));
    Console.WriteLine(Sum(1, 2));
    Console.WriteLine(Sum(1, 2, 3));
}
```

Синтаксический сахар
CIL ->

```
private static void Main(string[] args)
{
    Console.WriteLine(Program.Sum(1, 2, 3, 4));
    Console.WriteLine(Program.Sum(1, 2, 0, 0));
    Console.WriteLine(Program.Sum(1, 2, 3, 0));
}
```

Способы передачи параметров

По значению
(по умолчанию)

По ссылке
(модификатор ref)

По ссылке для возврата
(модификатор out)

Как массив
(модификатор params)

Передача параметров для возврата

- В метод передаётся ссылка на параметр
- Для обозначения способа передачи используется модификатор `out`
- Параметром может выступать только переменная или поле объекта
- Параметр обязан быть инициализирован в методе

Пример

```
static void Funct(out int n)
{
    n = int.Parse(Console.ReadLine());
    Console.WriteLine(n);
}

static void Main(string[] args)
{
    int x;
    Funct(out x);
    Console.WriteLine(x);
    Console.Read();
}
```



Передача параметров как массива

- В метод передаётся массив, содержащий параметры
- Для обозначения способа передачи используется параметр `params`
- Метод может принимать не более одного такого параметра
- Параметр с модификатором `params` должен быть последним в списке параметров

Пример

```
static int? Max(params int[] par)
{
    if (par.Length == 0)
    {
        return null;
    }
    int max = par[0];
    for (int i = 1; i < par.Length; i++)
    {
        if (max < par[i])
        {
            max = par[i];
        }
    }
    return max;
}
```

```
static void Main(string[] args)
{
    int? max = Max(3, 7, 4);
    if (max.HasValue)
    {
        Console.WriteLine(max);
    }
    else
    {
        Console.WriteLine("В метод не передано ни  
одного значения");
    }
    Console.Read();
}
```

Критерии создания методов

Принцип единственной ответственности
(1 метод – 1 действие)

Отсутствие повторяющихся фрагментов кода

UI и логика – это разные уровни действий

Размер метода не должен превышать 1-2 экранов

СПАСИБО ЗА ВНИМАНИЕ