

# Массивы в С#

Лекция №5

# Объявление массива

Массив – это последовательная группа переменных, имеющих одно имя и один тип.

Объявление массива имеет следующий вид:

```
<тип переменной> [] <имя массива> =  
new <тип переменной> [размерность  
массива]
```

# Объявление массива

Пример:

```
int [] a = new int[10] // объявлен массив a с  
размерностью 10
```

```
double [] b = new double[15] // объявлен массив  
b с размерностью 15
```

# Индексы массива

Индекс (порядковый номер) элемента массива всегда является целым числом. Отсчет начинается с 0.

Например в массиве целых чисел  $a$  с размерностью 14:

$a[0]$  – первый элемент массива

$a[13]$  – последний элемент массива

# Инициализация массива

Возможна инициализация массива.

Пример:

```
char [] charray = {'5', 'g', ';', '&', 'ю'};
```

```
char ch1 = charray[0]; // '5'
```

```
char ch3 = charray[3]; // '&'
```

# Генерация элементов массива

Для генерации элементов массива в C# используется класс `Random`. Чтобы им воспользоваться необходимо создать новый объект класса.

```
Random <имя объекта> = new Random();
```

# Генерация элементов массива

Элементы массива генерируются внутри цикла следующим образом:

```
for (int i=0; i<n; i++)  
{  
  a[i] = rn.Next(11);  
}
```

Next – это метод класса Random, который возвращает неотрицательное случайное целое число.

# Основные свойства и методы

Length (свойство класса Array) - общее число элементов во всех измерениях массива Array.

**<имя массива>.Length**

IndexOf и LastIndexOf (методы класса Array) - определяют индексы первого и последнего вхождения образца в массив, возвращая -1, если такового вхождения не обнаружено.

Применяется только для одномерного массива.

**Array.IndexOf(<имя массива>, <образец>)**

# Основные свойства и методы

Sort (метод класса Array) - осуществляет сортировку массива.

**Array.Sort(<имя массива>);**

Reverse (метод класса Array) - выполняет обращение одномерного массива, переставляя элементы в обратном порядке.

**Array.Reverse(<имя массива>, <образец>)**

# Оператор цикла foreach

Оператор цикла foreach перебирает подряд все элементы массива.

**foreach (<тип переменной>  
<имя переменной> in <имя массива>) {...}**

Тип переменной должен совпадать с типом элементов массива.

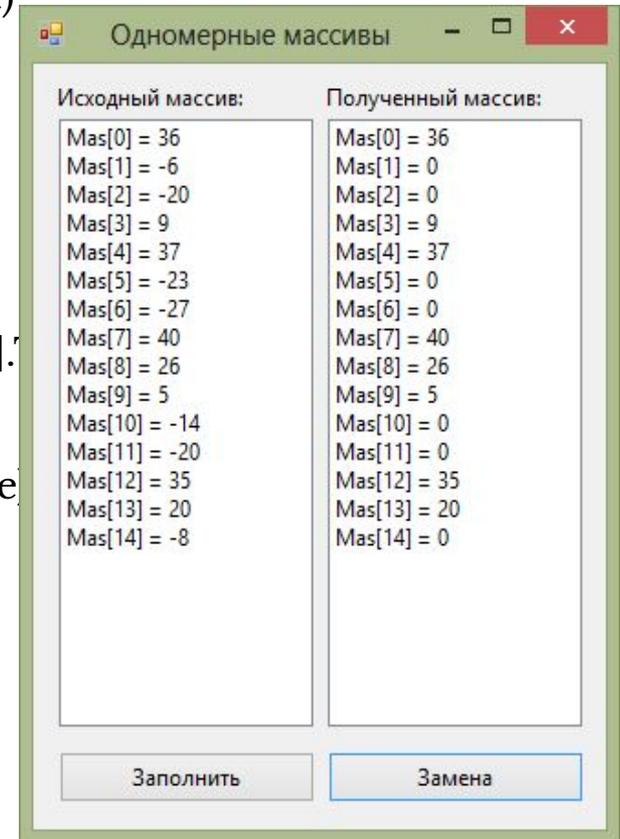
Пример:

```
int[] a = new int[5];  
foreach (int i in a) {...}
```

Создайте форму с элементами управления, как показано на рис. 7.1. Опишите одномерный массив. Создайте обработчики события для кно- пок (код приведен ниже). Данная программа заменяет все отрицатель- ные числа нулями. Протестируйте правильность выполнения програм- мы. Модифицируйте программу в соответствии с индивидуальным заданием.

```
int[] Mas = new int[15];
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    Random rand = new Random();
    for (int i = 0; i < 15; i++)
    {
        Mas[i] = rand.Next(-50, 50);

        listBox1.Items.Add("Mas[" + i.ToString() + "] = " + Mas[i].ToString());
    }
}
private void button2_Click(object sender, EventArgs e)
{
    listBox2.Items.Clear();
    for (int i = 0; i < 15; i++)
    {
        if (Mas[i] < 0)
            Mas[i] = 0;
        listBox2.Items.Add("Mas[" + Convert.ToString(i)
        + "] = " + Mas[i].ToString());
    }
}
```



# Вариант 1

- Дан целочисленный массив состоящий из 10 элементов. Элементы массива принимают значения в интервале  $[-7; 7]$ . Найдите последнее вхождение в массив элемента  $n$  ( $n$  вводится с клавиатуры). Подсчитайте количество элементов массива меньших  $n$ .

# Вариант 2

- Дан целочисленный массив состоящий из  $n$  элементов. Элементы массива принимают значения в интервале  $[-10; 10]$ . Найдите максимальный элемент массива