



# Программирование на C#

## Семинар №10

Модуль №1

Тема:

Индексные массивы

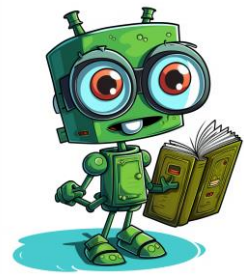
Массивы и методы: передача и возврат

Перегруженные методы



# Задания преподавателя к семинару

- Выполняем задания категорий ToDo и Self
- Методы для работы размещаем в отдельном статическом классе, класс в отдельном файле
- Не забываем контролировать корректность ввода данных и организовывать повторение решения задач



# Полезные материалы к семинару

1. Массивы в C#: <https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/>

# Demo 01. Массивы и Методы. Определить у какого из двух массивов максимальный элемент имеет большее значение.



Если массив передается как параметр метода, то использование `ref` не требуется:

```
static int MaxArray(int[] Arr)
{
    int maxValue;

    maxValue = Arr[0];
    for (int i=1; i<Arr.Length; i++)
    {
        if (Arr[i] > maxValue)
            { maxValue = Arr[i];}
    }
    return maxValue;
}
```

```
static void Main(string[] args)
{
    int[] Array1 = new int[5] {1,2,3,24,5};
    int[] Array2 = new int[5] {1,2,3,24,-5};

    if (MaxArray(Array1) > MaxArray(Array2) )
        {Console.WriteLine(" 1 > 2 ");}
    else
        if (MaxArray(Array1) <
            MaxArray(Array2))
            {Console.WriteLine(" 2 > 1 ");}
        else
            {Console.WriteLine(" 1 = 2 ");}
    Console.ReadKey();
}
```

# ToDo 01. Задание к Демо 01.



1. Дополните программу методами заполнения массива
2. Замените инициализацию массива на вызов метода заполнения массива.
3. Дополните программу методом вывода массива на экран и примените его.

# Оглавление (индексный массив) [index array] –индексов элементов некоторой последовательности



```
int len = lines.Length;
int[] index = new int[len]; // Массив индексов.
for (int i = 0; i < len; i++)
    { index[i] = i; }

// Сортировка массива индексов:
for (int i = 0; i < len - 1; i++)
    for (int j = i + 1; j < len; j++)
        if (lines[index[i]].Length > lines[index[j]].Length) {
            int temp = index[i];
            index[i] = index[j];
            index[j] = temp;
        }
//Вывод массива.
Console.WriteLine("Результат перебора:");
foreach (int n in index)
    { Console.WriteLine(lines[n] + " ");}
```

# Self. Задачи



**Self 01.** Создайте массив вещественных чисел и заполните при помощи датчика случайных чисел в диапазоне  $[-12.25 ; 12.25]$ . Заполнение массива и вывод на экран оформить в виде методов.

**Self 02.** Вспомните, как формируются перегруженные методы. Сформируйте перегруженные методы:

## Fill\_Array

- заполнение одномерного массива целых чисел с клавиатуры
- заполнение одномерного массива целых чисел случайными значениями
- заполнение одномерного массива вещественных чисел с клавиатуры
- заполнение одномерного массива вещественных чисел случайными значениями

## Print\_Array

- Вывод одномерного массива целых чисел
- Вывод одномерного массива вещественных чисел

Методы разместить в отдельном файле .cs.

Применить перегруженные методы

## Self. Задания для самостоятельной работы



**Self 03.** Напишите метод для нахождения **максимального элемента в одномерном массиве**. Метод должен возвращать значение **false**, если переданный массив пуст или равен null или **true** при удачном нахождении максимума.

Сам максимальный элемент должен сохраняться в виде out-параметра, при возврате false ему присваивается 0.

В основной программе организуйте ввод одномерного количества – первым указывается количество элементов, а затем сами элементы. После этого найдите максимальный элемент с помощью реализованного метода.

Обеспечьте проверку некорректных данных (в том числе на отрицательный размер массива).

Заголовок метода:

```
public static bool TryFindMaxElement(int[] array, out int maxValue)
```



# Self. Задания для самостоятельной работы



**Self 04.** Напишите метод, получающий на вход ссылки на два массива целых чисел и **добавляющий все чётные элементы первого массива в конец второго**. Если первый массив null или пустой, второй массив не должен меняться, а если второй массив пустой или null – по необходимости выделите память для хранения элементов.

В основной программе организуйте ввод размеров обоих массивов и проверку корректности значений, заполните массивы случайными числами в диапазоне  $[-100; 100]$ , а затем вызовите написанный метод. Последним шагом выведите оба массива, первый из них не должен измениться в результате вызова.

Подумайте, как должен выглядеть корректный заголовок метода, чтобы обеспечить изменение элементов.

## Self. Задания для самостоятельной работы



**Self 05.** Напишите метод, удаляющий все отрицательные элементы массива чисел типа **int**. Исходный массив должен уменьшаться до оставшегося количества элементов. Дополнительно метод возвращает новый массив, содержащий все удалённые элементы. Если не было удалено ни одного элемента, этот массив должен быть пустым.

В основной программе введите размер массива и его элементы, затем вызовите реализованный метод, выведите массив, полученный в результате вызова, на экран, а затем на отдельной строке выведите массив удалённых элементов.

Самостоятельно определите, какие из методов класса `Array` будут полезны для реализации.

# Self. Задания для самостоятельной работы



**Self 06.** Напишите метод для «сжатия массива» по правилу: если сумма двух соседних элементов  $A[i]$ ,  $A[i+1]$  делится на три, вычисляется их произведение и присваивается элементу  $A[i]$ . Элемент  $A[i+1]$  при этом удаляется, а элементы справа сдвигаются на одну позицию влево (размер массива при этом уменьшается на 1). После этого обработка элементов продолжается, начиная с  $(i + 1)$ -го и так до конца массива. Метод работает с исходным массивом, без дополнительных и буферных копирований.

Также напишите метод, последовательно вызывающий метод «сжатия массива» и возвращающий количество успешно проведенных операций до момента, когда дальнейшие сжатия перестанут изменять массив.

В основной программе вводится размер  $N$  целочисленного массива, создаётся массив, его элементам присваиваются случайные значения из диапазона  $[-100; 100]$ . Выведите исходный массив, после чего вызовите метод для него многократного сжатия, выведите количество сжатий и итоговый полученный массив.