

# Модуль 1

## Практическое занятие 5

**Массивы.**  
**Методы класса Array**  
**Оператор foreach**

# Задача 1

- В программе получить от пользователя количества элементов в двух целочисленных массивах: A и B. Массивы заполнить значениями случайными числами из диапазона [10, 50].
- Изменить массив A, добавив в конец все четные по значению элементы массива B.
- На экран вывести:
  - Исходный массив A
  - Исходный массив B
  - Модифицированный массив A
- **Решение основано на однократном проходе массивов и выделении памяти под массив A с запасом**

# Задача 1

```
int numOfItemsA, // количество элементов массива A
    numOfItemsB, // количество элементов массива B
    min = 10,    // минимум диапазона значений элементов
    max = 50;    // максимум диапазона значений элементов

do Console.Write("Введите размер массива A: ");
while (!int.TryParse(Console.ReadLine(), out numOfItemsA) ||
        numOfItemsA <= 0);

do Console.Write("Введите размер массива B: ");
while (!int.TryParse(Console.ReadLine(), out numOfItemsB) ||
        numOfItemsB <= 0);

int[] a = new int[numOfItemsA + numOfItemsB];
int[] b = new int[numOfItemsB];
```

# Задача 1

```
// Заполняем массивы случайными числами
Random generator = new Random();
for (int i = 0; i < numItemsA; i++) {
    a[i] = generator.Next(min, max + 1);
    Console.Write("{0,4}", a[i]); // и выводим сразу значения элементов
}
Console.WriteLine();

//TODO1: Самостоятельно заполнить случайными числами массив B
Console.WriteLine();
int k = numItemsA; // число элементов в дополненном массиве A
for (int i = 0; i < b.Length; i++)
    if (b[i] % 2 == 0) a[k++] = b[i];
// здесь можно создать новый массив с k элементами из A
int[] result = new int[k];
Array.Copy(a, result, k); // копирование из a в result k элементов

for (int i = 0; i < k; i++)
    Console.Write("{0,4}", a[i]);
```

## Задача 2

Получить от пользователя число **K** - размер массива и определить массив символов из **K** элементов.

Присвоить элементам массива случайные значения от 'A' до 'Z'.

Вывести полученный массив, затем создать копию массива, отсортировать элементы копии в алфавитном порядке и вывести.

Затем реверсировать (заменить порядок следования элементов на обратный) отсортированный массив и вывести на экран.

## Задача 2

```
char[] line; // массив символов
int K;       // количество элементов в массиве line
Random generator = new Random();

do Console.Write("Введите размер массива: ");
while (!int.TryParse(Console.ReadLine(), out K) || K <= 0);
line = new char[K];

for (int i = 0; i < K; i++) {
    line[i] = (char)generator.Next('A', 'Z' + 1);
    Console.Write("{0,2}", line[i]);
} //i
```

## Задача 2

```
Console.WriteLine();  
char[] newline = (char[])line.Clone();  
// упорядочение букв в массиве и вывод  
Array.Sort(newline);  
foreach (char ch in newline)  
    Console.Write("{0,2}", ch);  
Console.WriteLine();  
// инверсия порядка значений в массиве и вывод  
Array.Reverse(newline);  
foreach (char ch in newline)  
    Console.Write("{0,2}", ch);
```

## Задача 3

**3.1.** Написать метод, формирующий и возвращающий массив из **N** членов разложения в ряд функции **sin(1)**:

$$\sin(x) = x - x^3/3! + x^5/5! - \dots$$

Параметр **N** – число требуемых членов ряда.

**3.2.** Написать метод для вычисления **sin(x)** для заданного **x** с использованием массива членов ряда **sin(1)**.

Параметры: ссылка на массив разложения **sin(1)** и аргумент **x**.

**3.3.** В основной программе ввести значение **N** вычислить массив **sin(1)**. Вводя, последовательно вводя, значения **x**, вычислять **sin(x)** как с помощью созданного метода, так и с использованием библиотечного метода **Math.Sin()**. Сравнить результаты.

Вычисление **sin(x)** с точностью до машинного нуля см. на следующих слайдах

$$\sin(x) = x - x^3/3! + x^5/5! - \dots$$



## Задача 3

```
1.  // Переменные программы:
2.  double angle,    // Введенный угол в радианах
3.      x,           // Аргумент x (приведенный угол)
4.      sin,         // Сумма ряда (текущая)
5.      sinOld,      // Сумма ряда (предыдущая)
6.      memb;        // Очередной член ряда
7.                               // Ввод значения angle . . . .
8.      x = angle % (2 * Math.PI);    // - приведение угла
9.                               // Цикл вычисления суммы ряда (след. слайд)
10.                               // Вывод результатов (angle, sin)
11.                               // Вывод значения Math.Sin(angle)
12.                               // Завершение кода программы
```

## Задача 3

```
1.  int m; // число просуммированных членов ряда
2.  for (m = 1, sin = memb = x, sinOld = 0; sin != sinOld; m++) {
3.      Console.WriteLine("sin({0})={1} \tmemb = {2}", x, sin, memb);
4.      sinOld = sin;
5.      memb *= -x * x / 2 / m / (2 * m + 1);
6.      sin += memb;
7.  }
8.  //..Обратите внимание на вычисление  $x^n/n!$ 
```

## Задача 4

**4.1.** Написать метод для формирования массива, целочисленные неотрицательные элементы которого вычисляются по следующему рекуррентному соотношению:  $a_{i+1} = a_i \% 2 == 0 ? a_i / 2 : (3 * a_i + 1)$ .

Процесс вычисления завершается, когда  $a_{i+1}$  станет равным **1**.

Параметр метода задает значение  $a_0$ . Количество элементов массива заранее не известно, так как определяется значением  $a_0$ .

**4.2.** Написать метод вывода элементов массива в виде **[индекс] = значение**, размещая в строке по 5 элементов.

**4.3** В основной программе, вводя значения  $a_0$ , формировать массив и выводить его элементы на экран.

## Задача 5

"Сжать" массив, удалив из него отрицательные элементы

// Объявление массива с инициализацией:

```
int[] decArr = { -3, 7, 16, 10, -8, 0, 99, -64, 1, 9 };
```

```
int m = 0; // размер нового массива
```

```
for (int k = 0; k < decArr.Length; k++)
```

```
    if (decArr[k] >= 0) decArr[m++] = decArr[k];
```

```
if (m > 0) Array.Resize(ref decArr, m);
```

### Задание:

В основной программе ввести размер **N** целочисленного массива, создать массив, присвоить его элементам случайные значения из диапазона  $[-10;10]$ . Выполнить сжатие массива, удалив из него элементы с четными значениями.

## Задача 6

**6.1.** Написать метод, реализующий операцию «сжатия массива»: если сумма двух соседних элементов **A[i]**, **A[i+1]** начиная с первого делится на три, вычислить их произведение. Произведение присвоить элементу **A[i]**. Элемент **A[i+1]** удалить, элементы справа от **i** сдвигаются на одну позицию влево, размер массива уменьшается на **1**. Далее обрабатываются элементы, начиная с **(i+1)**-го и так до конца массива.

**6.2.** Написать метод, последовательно «запускающий» операцию (метод) сжатия массива и возвращающий количество успешно проведенных операций до момента, когда сжать массив больше невозможно.

В основной программе ввести размер **N** целочисленного массива, создать массив, присвоить его элементам случайные значения из диапазона **[-10;10]**. Выполнить сжатие массива.

## Задача 7

Написать метод для формирования массива символов, представляющих отдельные цифры целого числа ***N***, переданного в параметре

В основной программе, вводя значения целого ***N*** > 0, формировать символьный массив его цифр и выводить его элементы на экран.

# Задача 7

```
public static char[] GetDigits(int number) {  
    int len = (int)Math.Log10(number) + 1; // количество цифр числа  
    char[] digits = new char[len]; // массив для хранения цифр  
    int figure, i = len - 1;  
    do { // выделяем цифры числа и помещаем в массив  
        figure = number % 10;  
        number = number / 10;  
        digits[i--] = (char)(figure + '0');  
    }  
    while (number != 0);  
    return digits;  
}
```

```
int number;  
do Console.Write("Введите целое положительное число: ");  
while (!int.TryParse(Console.ReadLine(), out number) || number <= 0);  
char[] ciphers = GetDigits(number);  
Console.Write("Цифры числа: ");  
foreach (char ch in ciphers)  
    Console.Write(" " + ch);  
Console.WriteLine();
```

*Код метода Main()...*

## Задача 8

**8.1.** Написать метод формирования массива из ***N*** вещественных элементов, значения элементов массива вычисляются по формуле:

$$a_i = \frac{i(i+1)}{2} \% N; \quad i = 0, 1, 2, 3, \dots, (N-1)$$

**8.2.** Написать метод нормировки элементов вещественного массива путем деления каждого элемента на значение максимального по модулю элемента.

**8.3.** Написать метод вывода на экран значений элементов вещественного массива с точностью до трёх знаков после запятой.

**8.4.** В основной программе получать от пользователя значение ***N***, формировать и нормировать массив. Исходный и нормированный массивы выводить на экран.



# Задачи для самостоятельного решения

## Задача 9

9.1. Написать метод замены всех вхождений максимального элемента массива значением, переданным в параметре.

9.2. В основной программе объявить и инициализировать массив; получить от пользователя число, заменить им все вхождения максимального элемента в массив. Исходный и изменённый массивы вывести на экран.



*Microsoft Visual C# 2010: An introduction to object-oriented programming. 4-th edition, Chapter 6 до Multidimensional Arrays, прочитать, реализовать примеры, решить задачи.*



# Задачи для самостоятельного решения

1. Пользователем с клавиатуры вводится целое число  $N > 0$ . В программе сформировать и вывести на экран целочисленный массив из  $N$  элементов. Значениями элементов являются степени числа 2 от нулевой до  $N-1$  ( $1, 2, \dots, 2^{N-1}$ ). Заполнение массива степенями числа 2 организовать при помощи метода.
2. Пользователем с клавиатуры вводится целое число  $N > 0$ . В программе сформировать и вывести на экран целочисленный массив из  $N$  элементов, элементами которого являются нечетные числа от 1. Заполнение массива нечётными числами организовать при помощи метода.
3. Пользователем с клавиатуры вводятся целые числа  $N > 1$ ,  $A$  и  $D$ . В программе сформировать и вывести на экран целочисленный массив из  $N$  элементов. Элементы вычисляются:  $A[0] = A$ ,  $A[1] = A + D$ ,  $A[2] = A + 2 * D$ , ...  $A[N-1] = A + (N-1) * D$ . Формирование массива организовать при помощи метода.
4. Пользователем с клавиатуры вводится целое число  $N > 1$ . В программе сформировать целочисленный массив, содержащий  $N$  первых элементов последовательности Фибоначчи:  $A[0] = 1$ ,  $A[1] = 1$ ,  $A[2] = A[0] + A[1]$ , ...  $A[K] = A[K-1] + A[K-2]$ , ... Формирование массива организовать при помощи метода. Элементы массива вывести на экран в обратном порядке, методы класса **Array** не использовать.