Программирование на C#

Семинар №9

Модуль №1

Тема:

Одномерные массивы.

Инициализация элементов массива.

Генератор случайных чисел. Класс Random.

Методы класса Array.

Полезные материалы к семинару

1. Массивы в C#: https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/
2. Индексация с конца и диапазоны: https://learn.microsoft.com/en-us/dotnet/csharp/whats-new/tutorials/ranges-indexes
3. Класс Array. Методы https://learn.microsoft.com/ru-ru/dotnet/api/system.array?view=net-7.0
4. Методы класса Array:

- Основы: https://learn.microsoft.com/en-us/dotnet/api/system.array
- Метод Resize: https://learn.microsoft.com/en-us/dotnet/api/system.array.resize
- Метод Reverse: https://learn.microsoft.com/en-us/dotnet/api/system.array.reverse
- Метод Sort: https://learn.microsoft.com/en-us/dotnet/api/system.array.sort

5. Генерация случайных чисел:

Random: https://learn.microsoft.com/en-us/dotnet/api/system.random

Криптографически стойкий генератор случайных чисел (доп. материал): https://learn.microsoft.com/en-us/dotnet/api/system.security.cryptography.randomnumbergenerator

Задания преподавателя к семинару

- Выполняем задания категорий ToDo и Self
- Не забываем контролировать корректность ввода данных и организовывать повторение решения задач
- Начинаем выполнение заданий на базовые действия с массивами на слайде 12. Если вы уже знакомы с массивами в синтаксисе C, C++, C#, можете переходить к более сложным заданиям на самостоятельную работу.
- **Обязательными действиями** при работе с массивами являются:

- проверка корректности размерности массива

- организация вывода с форматированием

- вывод массива после заполнения и после каждого произведенного над ним действия

Одномерные массивы (векторы)

Этапы работы с массивами:

1. Объявление ссылки на массив: `int[ ] intArray;`
2. Создание экземпляра массива: `intArray = new int[100];`
3. Инициализация массива:

Явная инициализация без **new int[ ]:**

```
int[] arr2 = { 1, 2, 3 };
```

Явная инициализация, обязательно указать ровно 4 элемента в фигурных скобках:

```
int[] arr3 = new int[4] { 4, 5, 6, 7 };
```

Явная инициализация, размер массива определяется автоматически по фактическому размеру инициализатора, квадратные скобки пустые:

```
int[] arr4 = new int[] { 1, 11, 111 };
```

Явная инициализация, тип массива определяется автоматически компилятором по типу инициализатора:

```
var arr5 = new[] { 8, 9, 10 };
```

Элемент массивов может быть результатом вычислений выражения:

```
int[] arr6 = {(int)Math.Round(3.14), 20 + 5 };
```

Индексации в Цикле

```
//Классический цикл: например, for.

    int sum = 0;
    for (int i = 1; i < array.Length; i++)
          sum+=array[i];
```

NB! Тип элемента должен соответствовать типу элементов массива

Цикл работы с массивами:

```
        foreach (тип_Имя in Ссылка_на_массив)
```

```
    string[ ] S; // Ссылка на массив строк.
    foreach (string str in S)
       Console.WriteLine(str)
```

В строку str подгружаются по очереди все строки массива

Индексация с Конца. Index

В C# 8.0 была добавлена операция **^x** для получения **индекса с конца**.
NB!: Для последовательности длины length индекс с конца ^x будет вычисляться как **Length – x**. Таким образом, при индексации с конца последний элемент имеет **индекс ^1**, а первый – **^Length**.

```
int[] arr = { 1, 2, 3, 4, 5 };
Console.WriteLine(arr[^1]);     // 5.
Console.WriteLine(arr[^5]);     // 1.
```

```
int[] arr = { 35, 35, 555, 800, 8 };
// Обход массива в обратную сторону с помощью индексов с конца.
for (int i = 1; i <= arr.Length; i++)
{
    Console.Write(arr[^i] + " ");
}
```

Результат выполнения:
8 800 555 35 35

Диапазоны. Range

```
Range rng1 = ..;       // Эквивалент диапазона [0;^0).
Range rng2 = x..;      // Эквивалент диапазона [x;^0).
Range rng2 = ..y;      // Эквивалент диапазона [0;y).
```

```
int[] arr = { 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 };
int[] newArr1 = arr[5..11];

// Изменения старого массива не влияют на новый - он хранит копии значений.
arr[^4] = 100000;
```

```
foreach (int val in newArr1)
{
    Console.Write(val + " ");
}
```

Цикл foreach позволяет просматривать массивы: из массива newArr1 поочередно подгружаются элементы в переменную val (переменная того же типа, что и элементы массива

Методы класса Random

Генерация случайных положительных чисел.

1. **Создание экземпляра класса Random**: `Random rnd = new Random();`

При создании экземпляра в конструктор класса Random передается стартовое число (начальное значение):

Явно: Random(Int32)

По умолчанию: Random(). В платформа .NET Framework начальное значение по умолчанию зависит от времени. В .NET Core начальное значение по умолчанию создается поток-статическим генератором псевдослучайных чисел

2. **Применение методов класса Random**:

Next()	Возвращает неотрицательное случайное целое число.
Next(Int32)	Возвращает неотрицательное случайное целое число, которое меньше указанного максимального значения.
Next(Int32, Int32)	Возвращает случайное целое число в указанном диапазоне.

Random класс https://learn.microsoft.com/ruru/dotnet/api/system.random?view=net-7.0
Random конструкторы https://learn.microsoft.com/ru-ru/dotnet/api/system.random.-ctor?view=net-7.0

Demo 01. Класс Random

1. Сначала создаем экземпляр класса

```
Random rnd = new Random();
        int lowerBound = int.Parse(Console.ReadLine());
        int upperBound = int.Parse(Console.ReadLine());
        int arraySize  = int.Parse(Console.ReadLine());

        int[] numbers = new int[arraySize];
        for (int i = 0; i < numbers.Length; i++)
        {
            numbers[i] = rnd.Next(lowerBound, upperBound + 1);
        }
```

Свойство Length хранит длину массива

2. Применяем метод класса Random

Помните: верхняя граница указывается не включительно!

Подробнее о Random: https://learn.microsoft.com/en-us/dotnet/api/system.random

Класс Random.Генерация Вещественных Чисел

Метод **NextDouble ()** возвращает число типа double с плавающей запятой в диапазоне от 0.0 до 1.0.

```
Random rnd = new Random();
double number = rnd.NextDouble();
```

ToDo 01. Методы класса Random умеют генерировать отрицательные числа в ином диапазоне, чем NextDouble()? Реализуйте генерацию чисел в диапазоне от – 12.5 до 20.5

Demo 02. Методы Класса Array

```
int[ ] a,b; //Массивы a и b.
a=new int[ ]{10,20,30,40};
b=new int[a.Length]; // 2-ой массив той же длины.
Array.Copy (a, 1, b, 0, 2); // Копирование.
Array.Sort (b, 0, b.Length); // Отсортировали все элементы из b.
Array.Reverse (b, 1, 2);
```

```
//Массив a:                          10 20 30 40
//Массив b после копирования:        20 30  0  0
//Массив b после сортировки:          0  0 20 30
//Массив b после переворота:          0 20  0 30
```

Self. Задания На Базовые Действия с Массивами.

Выполняются на семинаре.

Self 01. Сформировать целочисленный массив A из N элементов при помощи ввода значений с клавиатуры (N вводит пользователь). Вывести массив на экран.

Self 02. Заменить в задаче Self 01 ввод на случайные числа в диапазоне [- 2 ; 7]

Self 03. Применить к ранее созданному массиву A метод сортировки класса Array. Далее отсортировать массив в порядке убывания и вывести на экран. Вспомните, какой метод класса Array может нам помочь?

Self 04. Из массива A копируем все четные элементы в массив B, нечетные – в массив C. Массивы выводим на экран.

Self. Задачи для самостоятельной работы

Self 05. Создать массив A из N положительных чисел. N ввести с клавиатуры. Элементы массива проинициализировать случайными числами в диапазоне [1,5]. Вывести на экран массив A. Если какое-либо число в массиве A повторяется, то каждое повторение заменить цифрой 0. В массиве A должны остаться только уникальные числа и нули. Вывести на экран модифицированный массив A.

Пример:

Исходный массив: 2, 5, 2, 3, 4, 5, 2, 4, 1 ,3 ,4 ,5

Модифицированный массив: 2, 5, 0, 3, 4, 0, 0, 0, 1, 0, 0, 0

Self 06. Дана строка, среди символов которой есть одна открывающаяся и одна закрывающаяся скобка. Вывести на экран все символы, расположенные внутри этих скобок подсчитать их кол-во.

Self 07. Создать массив целых чисел A из 20 элементов и инициализировать явно инициализатором. Массив целых чисел B из 10 элементов заполнить в диапазоне [-12 : 14]. Оба массива вывести на экран. В массив C скопировать элементы массива B и дописать в все четные элементы массива A, вывести на экран. Массив C должен иметь размерность, соответствующую количеству заносимых в него элементов.

Self. Задания для самостоятельной работы

Self 08. Пользователем с клавиатуры вводится целое число N > 0. В программе сформируйте и выведите целочисленный массив из N элементов. Значениями элементов – степени 2 от нулевой до ***N***-1 (1, 2, … 2^{N-1}).

Self 09. Пользователем с клавиатуры вводится целое число N > 0. В программе сформируйте и выведите целочисленный массив из N элементов, элементами которого являются нечётные числа от 1.

Self 10. Пользователем с клавиатуры вводятся целые числа N > 1, A и D. В программе сформируйте и выведите 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.

Self 11. Пользователем с клавиатуры вводится целое число N > 1. В программе сформируйте 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]