Материалы для подготовки к семинару 5, модуль 1

# Массив

Напомним, что массивы являются ссылочным типом данных.

имя\_переменной

значение\_адреса\_данных

значение\_данных

***Объекты ссылочных типов размещаются в «куче»***

**[managing heap]**

***Если ссылка не связана с данными, то ее значение - null***



# Работа с одномерным массивом

**имя\_ссылки\_на\_массив[индексирующее\_выражение]**

**тип[ ] имя\_ссылки**

***Операция индексирования***

***Объявление ссылки на массив***

Объявление ссылки на массив

Инициализация элементов массива

Работа с элементами массива

Создание экземпляра массива

***Создание экземпляра объекта конкретного типа***

**new тип[размер\_массива]**

Допустимо объединение

Например, ссылка на массив может быть объявлена так:

**int[] intArr; // ссылка на целочисленный массив**

**double[] arr; // ссылка на вещественный массив**

Допустимо совмещение описания ссылки с инициализацией массива списком элементов:

**int[] intArr = { 1, 3, -8, 19, 0, 11 };**

**double[] arr = { 3.14, 0.0, 0.5 };**

Или описание ссылки с последующей инициализацией элементов массива, так:

**int[] intArr; // ссылка**

**intArr = new int[5]; // выделение памяти**

**for (int i = 0; i < 5; i++) {**

**intArr[i] = i \* i - 1;**

**Console.Write(intArr[i] + " ");**

**}**

Или так:

**// ссылка**

**int[] intArr;**

**// выделение памяти**

**intArr = new int[5];**

**intArr[0] = -1;**

**intArr[1] = 12;**

**intArr[2] = 34;**

**intArr[3] = -23;**

**intArr[4] = 78;**

# Использование итератора foreach

**foreach** **(**тип идентификатор **in** ссылка**)**

**{**

<тело\_цикла>

**}**

**int[ ] arInt = { 22, 5, 12, 63, -6, -52, 77, 41, 35, 23 };**

**foreach (int memb in arInt)**

**Console.Write(memb + " ");**

**for (int i = 0; i < arInt.Length; i++)**

**Console.Write(arInt[i] + " ");**

**Сравните:**

По **memb** доступно только значение элемента

**arInt[i]** – доступ к значению

**i** – индекс элемента

Для работы с массивами в пространстве имён System языка C# существует класс Array. В нём содержатся реализации наиболее востребованных действий с массивами.

# Примеры

## Пример 1. Работа с массивом

1.1. Определить и инициализировать целочисленный массив из 10-ти элементов.

1.2. Упорядочить его элементы по возрастанию значений методом **Sort().**

1.3.Вывести измененный массив (применить оператор **foreach**).

1.4. Ввести целое число и методом двоичного поиска **BinarySearch()** найти номер элемента с этим значением в упорядоченном массиве.

1.5. Вывести результат поиска.

**using System;**

**public class Program {**

**public static void Main() {**

**int number; // число для поиска**

**// 1.1. Объявление с инициализацией:**

**int[] decArr = { -3, 7, 16, 10, -8, 99, 32, -64, 1, 0 };**

**// 1.2. упорядочение элементов по возрастанию**

**Array.Sort(decArr);**

**// 1.3. вывод изменённого массива**

**foreach (int item in decArr)**

**Console.Write("{0:d}\t", item);**

**// 1.4 ввод целого числа:**

**do Console.Write("Введите целое число: ");**

**while (!int.TryParse(Console.ReadLine(), out number));**

**// поиск числа в массиве:**

**int n = Array.BinarySearch(decArr, number);**

**// 1.5. вывод результата поиска**

**string outPut = String.Format("Значение {0} ", number);**

**outPut += n < 0 ? "не найдено" :**

**String.Format("находится на позиции {0}", n);**

**Console.WriteLine(outPut);**

**}**

**}**

## Пример 2. Передача массива в метод по ссылке

Программа должна считывать с клавиатуры данные от пользователя, пока он не введёт -1. Все положительные значения, вводимые пользователем должны попадать в целочисленный массив, отрицательные – отбрасываются. После завершения ввода в программе вычисляется процентное соотношение элементов с чётными и нечётными значениями. Элементы массива и статистика выводится на экран.

**using System;**

**public class Program {**

**// метод добавляет значение в конец массива**

**// массив передаём по ссылки, потому что внутри метода**

**// осуществлён вызов Array.Resize(), куда массив передаётся по**

**// ссылке.**

**// Если arr будет передан по значению, массив не будет изменён для**

**// кода, вызывающего addToEnd()**

**public static void addToEnd(ref int[] arr, int newVal) {**

**Array.Resize(ref arr, arr.Length + 1);**

**arr[arr.Length - 1] = newVal;**

**}**

**// в методе вычисляется процентное соотношения чётных и нечётных значений**

**public static void OddEvenStat(int[] arr, out double oddPer, out double evenPer) {**

**oddPer = 0.0;**

**evenPer = 0.0;**

**int oddsNumb=0, evenNumb=0;**

**foreach(int i in arr) {**

**if (i % 2 == 0) evenNumb++;**

**else oddsNumb++;**

**}**

**oddPer = oddsNumb \* arr.Length / 100.0;**

**evenPer = 100 - oddPer;**

**}**

**public static void Main() {**

**int testValue;**

**string input;**

**// переменная должна быть проинициализирована,**

**// иначе мы не сможем обрабатывать её в методе Resize()**

**int[] data = new int[0];**

**do {**

**Console.WriteLine("Введите целое неотрицательное число: ");**

**input = Console.ReadLine();**

**if (int.TryParse(input, out testValue) & testValue > 0)**

**addToEnd(ref data, testValue);**

**} while (testValue != -1);**

**foreach (int i in data) {**

**Console.Write(i + " ");**

**}**

**double odds, evens;**

**// получаем статистику**

**OddEvenStat(data, out odds, out evens);**

**Console.WriteLine("\nStatistics");**

**Console.WriteLine($"Odds: {odds:f2} Evens: {evens:f2}");**

**}**

**}**

# Статические классы и члены классов

Создать класс со статическими членами, константами и полями только для чтения. В отладочном режиме проследить (по F11) последовательность выполнения инициализаций и т.д.

**class myClassmate { // Одноклассник**

**const int apprenticeship = 4; // срок обучения (лет)**

**static int entranceYear = 2010; // год поступления**

**static myClassmate() {**

**entranceYear = 2014;**

**}**

**public myClassmate() { } // Конструктор умолчания**

**public myClassmate(string name, int by) {//Конструктор общего вида**

**this.name = name;**

**birthYear = by;**

**}**

**public string Information() { // Метод объекта**

**return "Фамилия: " + name + "; возраст: " +**

**(entranceYear - birthYear) +**

**" лет; год окончания: " +**

**(entranceYear + apprenticeship);**

**}**

**}**

**class Program {**

**static void Main( ) {**

**myClassmate Nan = new myClassmate();**

**Console.WriteLine(Nan.Information());**

**myClassmate Bob = new myClassmate("Смирнов", 1997);**

**Console.WriteLine(Bob.Information());**

**}**

**}**

**}**

# Задачи для подготовки к семинару 5

## Задача 1

Пользователем с клавиатуры вводится целые числа Nи ***K*** *(1 ≤* ***K*** *≤* ***N****).* Сформировать массив ***A***размера***N****.* Вывести элементы массива с индексами, кратными ***K****:* ***AK****,* ***A2·K****,* ***A3·K****, … .*

*Условный оператор не использовать.*

## Задача 2

Описать целочисленный массив из ***N*** элементов. ***N*** вводится с клавиатуры пользователем. Заполнить массив, используя формулу квадратных треугольных чисел. ***Sn* = 34·*Sn*-1 – *Sn*-2 + 2, *S*0 = 0, *S*1 = 1**. Элементы массива вывести на экран

## Задача 3

Описать метод **ArrayHill(A)**, возвращающий значение типа **void**, меняющий порядок следования элементов в вещественном массиве **A** на следующий: наименьший элемент располагается на первом месте, наименьший из оставшихся – на последнем, следующий по величине – на втором, следующий на предпоследнем и т.д. В вызывающей программе сформировать целочисленный массив размера **N** (вводится с клавиатуры пользователем), заполнить его случайными числами из диапазона [-10, 10] и обработать при помощи метода **ArrayHill()**. Исходные и результирующие значения элементов массива вывести на экран. Для формирования массива размера **N** и вывода массива на экран описать методы.

## Задача 4

Описать метод **ArrayItemDouble(A, X)**, возвращающий значение типа **void**, дублирующий в целочисленном массиве **A** элементы равные целочисленному значению переменной **X**. В вызывающей программе сформировать целочисленный массив размера **N** (вводится с клавиатуры пользователем), заполнить его случайными числами из диапазона [10, 100] и при помощи метода **ArrayItemDouble()** продублировать элементы равные введенному пользователем числу **Y**. Исходные и результирующие значения элементов массива вывести на экран. Для формирования массива размера **N** и вывода массива на экран описать методы.

# Полезные ссылки

1. Класс Array (<https://msdn.microsoft.com/ru-ru/library/system.array(v=vs.110).aspx>)
2. A pleasant new C# syntax for string interpolation (<http://www.informit.com/articles/article.aspx?p=2422807>)
3. String interpolation in C# 6 (<https://brosteins.com/2015/08/11/string-interpolation-in-c-6/>)