**Теория**

Массивы в Visual C# имеют следующие функции:

* Каждый элемент в массиве содержит значение.
* Первым элементом в массиве является элемент с индексом **0**.
* Размер массива — это общее количество элементов, которые он может содержать.
* Массивы могут быть одномерными и многомерными.

**ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ МАССИВОВ:**

**Объявление и создание массива**

В следующем примере кода показано, как создать одномерный массив целых чисел с элементами от **0** до **9**:

|  |
| --- |
| **int**[] arrayName = new **int**[10]; |

Создание и одновременно инициализация массива:

|  |
| --- |
| **int**[] b = { 1, 2, 3, 4, 5 }; |

В следующем примере кода используется индекс для доступа к элементу с индексом **2**:

|  |
| --- |
| **int**[] oldNumbers = { 1, 2, 3, 4, 5 };  **int** number = oldNumbers[2]; *// 3* |

В следующем примере кода показано, как использовать цикл for для итерации по массиву:

|  |
| --- |
| **int**[] oldNumbers = { 1, 2, 3, 4, 5 };  **for** (**int** i = 0; i < oldNumbers.Length; i++)  {  **int** number = oldNumbers[i];  ...  } |

Использование цикла foreach:

|  |
| --- |
| **foreach** (**var** x **in** oldNumbers )  Console.Write($"{x} "); |

**ФУНКЦИЯ RANDOM**

Рассмотрим пример инициализации массива из пяти элементов байтового типа со случайными значениями:  
**1.** Создайте экземпляр генератора случайных чисел, используя системное значение в качестве начального:

|  |
| --- |
| **var** rand = new Random(); |

**2.** Сгенерируйте и отобразите 5 случайных значений:

|  |
| --- |
| **var** bytes = new **byte**[5];  rand.NextBytes(bytes);  Console.WriteLine("Five random byte values:");  *// output:*  **foreach** (**byte** byteValue **in** bytes)  Console.Write("{0, 5}", byteValue);  Console.WriteLine(); |

**Задания и лабораторные си шарп**

**Лабораторная работа 1 (Lab 1). Работа с одномерными массивами**

**Выполнить:** Дан массив целых чисел (значения элементов: -1, -2, 3, 4, 5, 6, 7). Создайте функцию для вывода массива. Создайте еще одну функцию для вычисления числа нечетных элементов в этом массиве, а также числа положительных элементов в нем.

**Указание 1:** Создайте функцию Print для вывода элементов массива.  
**Указание 2:** Создайте функцию CountOddPositive с циклом foreach и операторами if, чтобы проверить, является ли элемент нечетным или положительным.

**Пример выполнения:**

Массив:

-1, -2, 3, 4, 5, 6, 7

кол-во нечетных элементов: 5, кол-во положительных 5.

[Название проекта: Lesson\_7Lab1, название файла L7Lab1.cs]

 Алгоритм:

1. Создайте консольное приложение с именем Lesson\_7Lab1.
2. В окне *Обозреватель решений* (Solution Explorer) найдите файл Program.cs и переименуйте его в L7Lab1.cs.
3. Подключите класс Console, чтобы постоянно не писать его название при обращении к его методам:

...

**using static** System.Console;

...

1. В функции Main() создайте одномерный целочисленный массив a и инициализируйте его элементы значениями:

...

**int**[] a = new **int**[] { -1, -2, 3, 4, 5, 6, 7 };

...

1. Ниже функции Main() (после ее закрывающей фигурной скобки) создайте метод Print с одним параметром arr — целочисленный массив — который будет принимать массив из основной программы. Метод не должен ничего возвращать (void):

static **void** Print(**int**[] arr)

{

WriteLine("Массив:");

foreach (var x in arr)

Write(x + " ");

WriteLine();

}

1. Вернитесь в функцию Main и вызовите метод для печати элементов массива. Запустите приложение.

|  |
| --- |
| ...  Print(a); |

1. Можно использовать сочетание клавиш Ctrl+M (дважды) чтобы временно свернуть или развернуть код метода.
2. Сейчас необходимо создать функцию CountOddPositive, которая принимает массив в качестве параметра и, кроме того, имеет еще два параметра *ref* (параметры ввода-вывода) для передачи значений счетчиков нечетных и положительных элементов (countPos и countOdd) :

...

static **void** CountOddPositive( **int**[] arr, ref **int** countOdd, ref **int** countPos)

{

}

...

1. Внутри метода лучше использовать цикл foreach для перебора и проверки элементов массива:

...

static void CountOddPositive( int[] arr, ref int countOdd, ref int countPos)

{

foreach (var x in arr)

{

if (x % 2 != 0)

countOdd++;

if (x > 0)

countPos++;

}

}

...

1. *Ref* параметр должен быть инициализирован до вызова метода. Давайте инициализируем параметр в функции Main:

...

**int** countPos = 0;

**int** countOdd = 0;

...

1. Теперь можно вызывать созданный метод CountOddPositive, передав ему массив в качестве первого аргумента:

...

CountOddPositive(a, ref countPos, ref countOdd);

...

1. Выведите результаты:

...

WriteLine($"кол-во нечетных элементов: {countOdd}, количество положительных элементов {countOdd}.");

...

1. Запустите программу CTRL+F5.

**Задание 1:**

**To do:** Дан массив вещественных чисел (значения элементов: 1.1, -2.3, 3.7, 4.1, 5.6, 6.1, 7.1).  
**1**. Создайте функцию для печати массива.  
**2**. Создайте еще одну функцию для поиска минимального и максимального элементов массива. При этом использовать стандартные min и max функции запрещено.  
    
**Указание 1:** Поиск максимального и минимального значения осуществлен в **Лабораторной работе 4** [5-го урока](https://labs-org.ru/c-sharp5/).  
   
**Указание 2:** Создайте функцию Print для печати элементов массива.  
   
**Указание 3:** Создайте функцию FindMaxMin с циклом *foreach* и оператором *if* для поиска максимального и минимального значения. Заголовок функции должен выглядеть следующим образом:

|  |
| --- |
| **static** **void** FindMaxMin(**double**[] arr, **ref** **double** max, **ref** **double** min)  {  } |

**Пример выполнения:**

Массив:

1.1 -2.3 3.7 4.1 5.6 6.1 7.1

максимальный элемент: 7.1, минимальный элемент: -2.3

[Название проекта: Lesson\_7Task1, название файла L7Task1.cs]

**Лабораторная работа 2 (Lab 2). Random**

**Выполнить:**  
**1**. Создайте функцию FillRandomArray для заполнения массива из **10** элементов случайно сгенерированными числами диапазона от **-10** до **15**.  
**2**. Создайте еще одну функцию PrintArray для вывода элементов массива.  
**3**. Создайте функцию DivisibleBy3 для подсчета и вывода количества элементов массива, кратных *3*.

**Пример выполнения:**

Массив:

-1, -2, 3, 4, 5, 6, 7, 12, 9, 10

элементы кратные 3:

3 6 12 9

кол-во элементов кратных 3: 4

[Название проекта: Lesson\_7Lab2, название файла L7Lab2.cs]

 Алгоритм:

1. Создайте приложение с именем Lesson\_7Lab2. Переименуйте файл Program.cs в L7Lab2.cs.
2. Подключите класс Console:

|  |
| --- |
| ...  **using** **static** System.Console;  ... |

1. В функции Main() объявите одномерный массив из 10 элементов:

...

**int**[] arr = new **int**[10];

...

1. Ниже функции Main() (после ее закрывающей фигурной скобки) создайте метод FillRandomArray с тремя параметрами: arr параметр-массив типа *int* для получения массива из основной программы, minValue параметр типа *int* для установления нижней границы генерируемых значений функции Random, и maxValue параметр типа *int* для установления верхней границы функции Random. Создаваемая пользовательская функция не должна возвращать значение (void):

...

static **void** FillRandomArray(**int**[] arr, **int** minValue = -10, **int** maxValue = 15)

{

//...

}

...

1. Внутри функции создайте экземпляр объекта Random:

Random rand = new Random();

1. Сгенерируйте значения элементов массива и выведите их, используя верхнюю и нижнюю границу диапазона (параметры minValue и maxValue) :

...

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

{

arr[i] = rand.Next(minValue, maxValue);

}

1. Вызовите метод (FillRandomArray) в теле функции Main, передав ей массив в качестве параметра, а также границы диапазона генерируемых значений: **-10** и **15** :

...

FillRandomArray(arr, -10, 15);

...

1. Создайте метод PrintArray с одним параметром типа целочисленный массив (arr). Метод не должен возвращать значение (void), а должен выводить в консоль значения элементов массива:

static void PrintArray(int[] arr)

{

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

{

Write($"{arr[i]} ");

}

}

1. Вызовите метод (PrintArray) в теле функции Main, передав массив в качестве аргумента:

PrintArray(arr);

1. Создайте функцию DivisibleBy3, которая ищет элементы, кратные *3*, и выводит их.

static **void** DivisibleBy3(**int**[] arr)

{

**int** counter = 0;

WriteLine("элементы, кратные 3:");

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

{

if (arr[i] % 3 == 0) {

Write($"{arr[i]} ");

counter++;

}

}

WriteLine();

WriteLine($"кол-во элементов кратных 3: {counter}");

}

1. Вызовите метод (DivisibleBy3) в теле функции Main, после вызовов двух предыдущих функций:

...

DivisibleBy3(arr);

1. Запустите приложение CTRL+F5.
2. Добавьте комментарий с текстом задания и загрузите файл .cs в *moodle* систему.

**Задание 2:**

**Выполнить:** Дан целочисленный массив (15 элементов, сгенерированных случайным образом в диапазоне от *1* до *10*). Требуется определить, есть ли в массиве элемент, равный N (N вводится). Выведите *«да»* или *«нет»* один раз. Для поиска элемента используйте цикл foreach.  
**1**. Для заполнения и вывода элементов массива требуется создать функцию.  
**2**. Создайте еще одну функцию FindN для поиска элемента, равного N (N — один из параметров функции).

**Указание 1:** Для выхода из цикла используйте оператор break:

|  |
| --- |
| **foreach** (**int** x **in** arr)  {  **if** (...)  {  ...  **break**; *// выход*  }  } |

**Указание 2:** Для проверки того, найден ли искомый элемент, используйте логическую переменную:

|  |
| --- |
| **bool** f = **false**; |

**Примерный результат:**

Массив:

9 5 1 3 7 7 8 1 4 4

введите искомое значение:

6

результат: нет

***Дополнительное задание:*** Выведите также индекс найденного элемента.

[Название проекта: Lesson\_7Task2, название файла L7Task2.cs]

**Задание 3:**

**Выполнить:** Задан массив вещественных чисел (double) (**10** элементов, сгенерированных случайным образом в диапазоне от -5.0 до 5.0).  
**1**. Создайте функцию для заполнения и вывода элементов массива.  
**2**. Создайте еще одну функцию для подсчета и печати суммы троек рядом стоящих элементов: a[1]+a[2]+a[3], a[2]+a[3]+a[4], a[3]+a[4]+a[5], …… , a[8]+a[9]+a[10].  
    
**Указание 1:** Для генерации случайных вещественных чисел используйте границы:

|  |
| --- |
| *// например, от -20 до 20:*  Random rand = new Random();  ...  a[i]=rand.NextDouble() \* 40 - 20; |

**Указание 2:** Для вывода вещественных значений с указанием количества цифр после десятичной точки:

|  |
| --- |
| *// две цифры после десятичной точки*  Write("{0:0.00} ",arr[i] ); *// например 1.21* |

**Пример выполнения:**

массив:

-3,54 0,51 3,79 1,85 -4,11 -2,32 -1,35 -0,78 0,63 4,31

суммы троек:

0,76 6,15 1,53 -4,59 -7,79 -4,46 -1,50 4,16

[Название проекта: Lesson\_7Task3, название файла L7Task3.cs]

**ПРИЛОЖЕНИЯ ДЛЯ WINDOWS FORMS**

**Лабораторная работа 3 (Lab 3)**

**Выполнить:** Создайте проект для подсчета суммы и среднего арифметического значения элементов одномерного массива.

**Пример выполнения:**

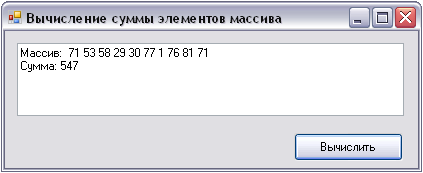
[](https://labs-org.ru/wp-content/uploads/2017/02/7-3.png)

Рис. Форма «Массивы»

[Название проекта: Lesson\_7Lab3, название файла L7Lab3.cs]

***Выполнение:***

1. Создайте новый проект. Расположите элементы управления на новой форме (см. рисунок).
2. Задайте свойству *Multiline* для текстового окна *txtArray* значение равное *true* (для того, чтобы в текстовом окне можно было выводить текст в несколько строк).
3. Далее необходимо запрограммировать кнопку *Вычислить* (btnCalc) так, чтобы в текстовое окно выводились элементы массива, их сумма, а затем их среднее арифметическое. Для этого в процедуре, описывающей событие щелчка мыши по кнопке *Вычислить*, опишем переменные, которые мы будем использовать при решении поставленной задачи:

|  |
| --- |
| **private** **void** btnCalc\_Click(**object** sender, EventArgs e)  {  **int**[] arr = new **int**[10];  **int** sum=0;  } |

1. Для среднего арифметического специальную переменную описывать не надо, потому что оно вычисляется по формуле *sum* разделить на *10*.
2. Для того чтобы задать значения элементов массива, воспользуемся циклом. Для генерации случайных чисел будем использовать переменную — экземпляр объекта Random:

|  |
| --- |
| Random rand = new Random();  **for**(**int** i=0;i<arr.Length;i++)  {  arr[i] = rand.Next(20);  *//...*  } |

1. Добавьте в цикл вычисление суммы элементов массива:

|  |
| --- |
| *//...*  sum += arr[i]; |

1. Теперь осталось вывести элементы массива, сумму и среднее арифметическое в текстовое окно.
2. Сначала выведите слово *Массив*:

|  |
| --- |
| txtArray.Text = "Массив: "; |

1. Теперь в цикле необходимо вывести элементы массива:

|  |
| --- |
| **for** (**int** i = 0; i < arr.Length; i++)  {  txtArray.Text += arr[i].ToString()+" ";  } |

1. Самостоятельно добавьте вывод в текстовое окно среднего арифметического.
2. Запустите и отладьте программу.

***Дополнительное задание:***

 Разработать приложение, которое находит максимальное и минимальное число из 15 элементов массива.

***Контрольное задание:***

 Добавьте в приложение возможность сортировки массива по возрастанию.

***Вопросы для самоконтроля:***

1. Как объявляется массив в VC#?
2. Сколько элементов будет содержать массив, который описан с помощью следующего оператора:

|  |
| --- |
| **int**[] arr = new **int**[5]; |

1. С помощью какого ключевого слова можно описать массив, который будет доступен всем модулям приложения?