

ПРАКТИЧЕСКАЯ РАБОТА № 6. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ОДНОМЕРНЫХ МАССИВОВ

Цель практической работы: Изучить способы получения случайных чисел. Написать программу для работы с одномерными массивами.

6.1. Работа с массивами

Массив - набор элементов одного и того же типа, объединенных общим именем. Массивы в C# можно использовать по аналогии с тем, как они используются в других языках программирования. Однако C#-массивы имеют существенные отличия: они относятся к ссылочным типам данных, более того - реализованы как объекты. Фактически имя массива является ссылкой на область кучи (динамической памяти), в которой последовательно размещается набор элементов определенного типа. Выделение памяти под элементы происходит на этапе инициализации массива. А за освобождением памяти следит система сборки мусора - неиспользуемые массивы автоматически утилизируются данной системой.

Рассмотрим в данной практической работе одномерные массивы. *Одномерный массив* - это фиксированное количество элементов одного и того же типа, объединенных общим именем, где каждый элемент имеет свой номер. Нумерация элементов массива в C# начинается с нуля, то есть, если массив состоит из 10 элементов, то его элементы будут иметь следующие номера: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Одномерный массив в C# реализуется как объект, поэтому его создание представляет собой двухступенчатый процесс. Сначала объявляется ссылочная переменная на массив, затем выделяется память под требуемое количество элементов базового типа, и ссылочной переменной присваивается адрес нулевого элемента в массиве. Базовый тип определяет тип данных каждого элемента массива. Количество элементов, которые будут храниться в массиве, определяется размер массива.

В общем случае процесс объявления переменной типа массив, и выделение необходимого объема памяти может быть разделено. Кроме того, на этапе объявления массива можно произвести его инициализацию. Поэтому для объявления одномерного массива может использоваться одна из следующих форм записи:

базовый_тип [] имя_массива;

Описана ссылка на одномерный массив, которая в дальнейшем может быть использована для адресации на уже существующий массив. Например:

int [] a;

базовый_тип [] имя_массива = new базовый_тип [размер];

Объявлен одномерный массив заданного типа и выделена память под одномерный массив указанной размерности. Адрес данной области памяти записан в ссылочную переменную. Элементы массива равны нулю (В C# элементам массива присваиваются начальные значения по умолчанию в зависимости от базового типа. Для арифметических типов - нули, для

ссылочных типов - null, для символов - пробел). Например: `int []a=new int [10];`

базовый_тип [] имя__массива={список инициализации};

Выделена память под одномерный массив, размерность которого соответствует количеству элементов в списке инициализации. Адрес этой области памяти записан в ссылочную переменную. Значение элементов массива соответствует списку инициализации. Например: `int []a={0, 1, 2, 3};`

Обращения к элементам массива происходит с помощью индекса, для этого нужно указать имя массива и в квадратных скобках его номер. Например, `a[0]`, `b[10]`, `c[i]`.

Так как массив представляет собой набор элементов, объединенных общим именем, то обработка массива обычно производится в цикле. Например:

```
int[] myArray = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
int i;
for (i = 0; i < 10; ++i)
    Console.WriteLine(myArray[i]);
```

6.2. Случайные числа

Одним из способов задания массива является задание определение элементов через случайные числа. Для работы со случайными числами используются в основном два метода класса **Random: Random и Next**. Метод **Random** подготавливает работу со случайными числами, обеспечивая, надежный способ создания непредсказуемой последовательности чисел.

Метод **Random.Next** создает случайное число в диапазоне значений от нуля до **Int32.MaxValue**. Для создания случайного числа в диапазоне от нуля до какого-либо другого положительного числа используется перегрузка метода **Random.Next(Int32)**. Для создания случайного числа в другом диапазоне используется перегрузка метода **Random.Next(Int32, Int32)**.

6.3. Порядок выполнения индивидуального задания

Создайте форму с элементами управления как приведено на рис. 6.1. Опишите одномерный массив. Создайте обработчики события для кнопок (код приведен ниже). Данная программа заменяет все отрицательные числа нулями. Протестируйте правильность выполнения программы. Модифицируйте программу в соответствии с индивидуальным заданием.

Рис. 6.1. Окно программы для работы с одномерными массивами

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        int[] Mas = new int[15];

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Random rand = new Random();
            textBox1.Text = "";
            for (int i = 0; i < 15; i++)
            {
                Mas[i] = rand.Next(-50, 50);
                textBox1.Text += "Mas[" + Convert.ToString(i) + "] = "
                    + Convert.ToString(Mas[i]) + Environment.NewLine;
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {

```

```

        textBox2.Text = "";
        for (int i = 0; i < 15; i++)
        {
            if (Mas[i] < 0) Mas[i] = 0;
            textBox2.Text += "Mas[" + Convert.ToString(i) + "] = "
                            + Convert.ToString(Mas[i]) + Environment.NewLine;
        }
    }
}

```

6.3. Варианты заданий

1. Найти сумму четных элементов массива.
2. Найти сумму положительных элементов массива.
3. Найти произведение элементов массива с четными индексами.
4. Найти количество нулевых и количество отрицательных элементов массива.
5. Найти сумму всех элементов массива, которые без остатка делятся на "6".
6. Найти количество положительных элементов массива, не превышающих числа "7".
7. Заменить отрицательные элементы нулями.
8. Увеличить все положительные элементы массива на единицу.
9. Найти произведение ненулевых элементов массива.
10. Увеличить элементы массива с четными индексами на "1", а элементы с нечетными индексами на "2".
11. Найти сумму отрицательных элементов массива а произведение положительных элементов.
12. Найти среднее арифметическое и среднее геометрическое значения элементов массива.
13. Указать номера элементов массива, которые без остатка делятся на "3" и на "5".
14. Указать номера элементов массива, которые больше "2", но меньше "5".
15. Указать, какой процент от общего количества составляют положительные элементы массива.
16. Сделать все положительные элементы массива отрицательными, вывести их индексы.