

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра программных систем

**Лабораторная работа №5**

**Создание и использование массивов**

Выполнил: Мороз

Илья Олегович

Группа № K3120

Проверила: Казанова П.П.

Санкт-Петербург

2019

**Цель работы:**

Целью работы является знакомство с массивами в языке C# и их применение на практике.

**Ход работы:**

**Упражнение 1.**

1. На основе проекта Loop из Лабораторной работы 3 была реализована работа с массивами.
2. Массив выводится в консоль с помощью цикла for. При этом чётные элементы массива заменяются на 0.

int[] myArray = { 100, 1, 32, 3, 14, 25, 6, 17, 8, 99 };

for (int j = 0; j < myArray.Length; j++)

{

if (myArray[j] % 2 == 0) myArray[j] = 0;

Console.Write(myArray[j] + " ");

}

Листинг 1 – работа с массивом

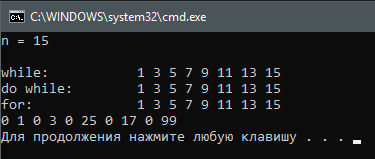


Рисунок 1 – результат работы программы

1. Была реализована работа с массивом, длина которого задаётся пользователем.
2. Вывод такого массива был реализован с помощью цикла foreach.

int[] myArray2;

Console.Write("n = ");

n = int.Parse(Console.ReadLine());

myArray2 = new int[n];

for (int j = 0; j < myArray2.Length; j++)

{

Console.Write("a[{0}] = ", j);

myArray2[j] = int.Parse(Console.ReadLine());

}

foreach (int x in myArray2)

{

Console.Write(x + " ");

}

Листинг 2 – работа с массивом с неизвестной длиной

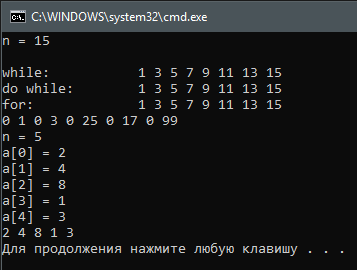


Рисунок 2 – результат работы программы

**Упражнение 2.**

1. Был создан новый проект MatrixMultiply.
2. Было реализовано заполнение двух двумерных массивов программно.
3. С помощью формул было произведено вычисление результата перемножения данных матриц.
4. С помощью рефакторинга нахождение результата, а также заполнение матриц было вынесено в отдельные методы.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MatrixMultiply

{

class MatrixMultiply

{

static void Main(string[] args)

{

int[,] a = new int[2, 2];

Input(a);

int[,] b = new int[2, 2];

Input(b);

int[,] result = Multiply(a, b);

Output(result);

}

private static void Input(int[,] a)

{

for (int r = 0; r < a.GetLength(0); r++)

{

for (int c = 0; c < a.GetLength(1); c++)

{

Console.Write("Enter value of [{0}, {1}]: ", r, c);

a[r, c] = int.Parse(Console.ReadLine());

}

}

Console.WriteLine();

}

private static int[,] Multiply(int[,] a, int[,] b)

{

int[,] result = new int[2, 2];

for (int r = 0; r < result.GetLength(0); r++)

{

for (int c = 0; c < result.GetLength(1); c++)

{

result[r, c] += a[r, 0] \* b[0, c] + a[r, 1] \* b[1, c];

}

}

return result;

}

private static void Output(int[,] result)

{

for (int r = 0; r < result.GetLength(0); r++)

{

for (int c = 0; c < result.GetLength(1); c++)

{

Console.Write("{0}\t", result[r, c]);

}

Console.WriteLine();

}

}

}

}

Листинг 3 – исходный код программы MatrixMultiply.cs

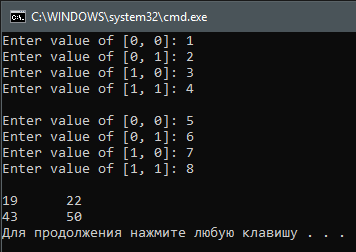


Рисунок 3 – результат работы программы

**Упражнение 3.**

1. Был создан новый проект FunWithArrays.
2. В программе были реализованы стандартные алгоритмы работы с массивами: нахождение суммы элементов, выборка элементов по индексу, нахождение максимального и минимального значений.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FunWithArrays

{

class Program

{

static void Main(string[] args)

{

int[] a;

int n = 1;

Console.Write("n = ");

n = int.Parse(Console.ReadLine());

a = new int[n];

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

{

Console.Write("a[{0}] = ", i);

a[i] = int.Parse(Console.ReadLine());

}

Console.WriteLine("Sum = {0}", GetSum(a));

Console.WriteLine("Average = {0}", GetAverage(a));

Console.WriteLine("Sum of positives = {0}", GetSumOfPositives(a));

Console.WriteLine("Sum of evenes = {0}", GetSumOfEvenes(a));

Console.WriteLine("Index of min = {0}", FindIndexOfMin(a));

Console.WriteLine("Index of max = {0}", FindIndexOfMax(a));

Console.WriteLine("Multiply from min to max = {0}", GetMultiplyFromMinToMax(a));

}

private static int GetSum(int[] a)

{

int sum = 0;

foreach (int x in a)

{

sum += x;

}

return sum;

}

private static double GetAverage(int[] a)

{

return (double)(GetSum(a) / a.Length);

}

private static int GetSumOfPositives(int[] a)

{

int sum = 0;

foreach (int x in a)

{

if (x > 0) sum += x;

}

return sum;

}

private static int GetSumOfEvenes(int[] a)

{

int sum = 0;

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

{

if (i % 2 == 0) sum += a[i];

}

return sum;

}

private static int FindIndexOfMin(int[] a)

{

int min = a[0];

int index = 0;

for (int i = 1; i < a.Length; i++)

{

if (a[i] < min)

{

min = a[i];

index = i;

}

}

return index;

}

private static int FindIndexOfMax(int[] a)

{

int max = a[0];

int index = 0;

for (int i = 1; i < a.Length; i++)

{

if (a[i] > max)

{

max = a[i];

index = i;

}

}

return index;

}

private static int GetMultiplyFromMinToMax(int[] a)

{

int multiply = 1;

int indexMin = FindIndexOfMin(a);

int indexMax = FindIndexOfMax(a);

if (indexMin > indexMax)

{

int temp = indexMin;

indexMin = indexMax;

indexMax = temp;

}

for (int i = indexMin; i <= indexMax; i++)

{

multiply \*= a[i];

}

return multiply;

}

}

}

Листинг 4 – исходный код программы FunWithArrays.cs

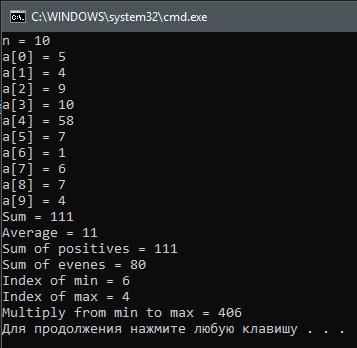


Рисунок 4 – пример работы программы

**Вывод:**

В ходе выполнения данной лабораторной работы были получены новые знания в языке C#. Были рассмотрены массивы, а также их возможности. Были рассмотрены алгоритмы работы с массивами, на основе которых была написана программа FunWithArrays.

**Ответы на контрольные вопросы:**

1. Что такое массив?

Это упорядоченная система элементов, каждый элемент имеет свой индекс.

1. Что такое константа?

Неизменяемая по значению область памяти.