Лабораторная работа №6

**Тема:** Разработка программ с использованием массивов.

**Цель:** Научится разрабатывать алгоритмы и реализовывать программы по обработке массивов с применением возможностей класса Array.

**Техническое оснащение:** OS Windows 10, MS VS 2019, MS Word.

Выполнение работы

Вариант 18

**Пример 1.** Дан массив из n действительных чисел. Вычесть из каждого элемента массива среднее значение массива. Распечатать полученный массив.

Результат работы примера (рисунок 1).

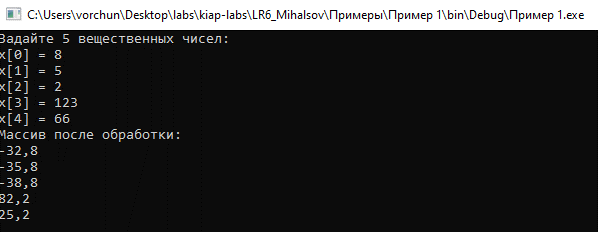


Рисунок 1 – Выполнение примера 1

**Пример 2.** Прономеровать матрицу действительных чисел A[2x3], т.е. каждый элемент матрицы поделить на максимальное по модулю число.

Результат работы примера (рисунок 2).

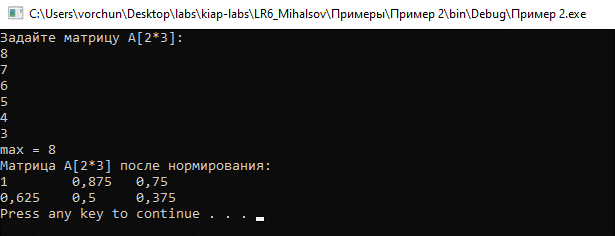


Рисунок 2 – Выполнение примера 2

**Пример 3.** Использование методов класса Array с одномерным массивом.

Результат работы примера (рисунок 3).

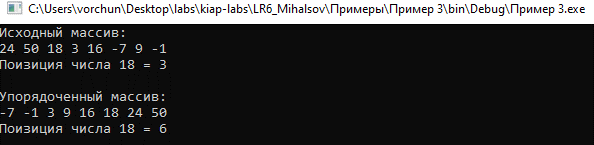


Рисунок 3 – Выполнение работы примера 3

**Пример 4.** Разработаем класс для работы с одномерным массивом. Создадим в нём индексатор, позволяющий обращаться к элементу массива по индексу.

Результат работы примера (рисунок 4).

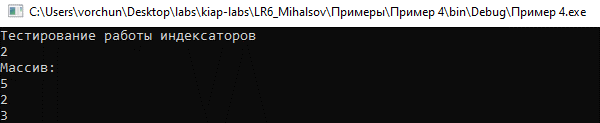


Рисунок 4 – Выполнение примера 4

**Пример 5.** Разработаем класс для работы с двумерным массивом (матрицей). Создадим в нём индексатор, позволяющий обращаться к элементу матрицы по индексу.

Результат работы примера (рисунок 5).

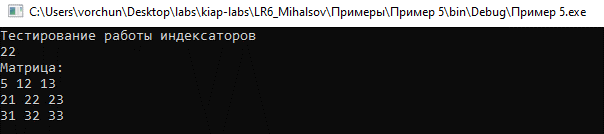


Рисунок 5 – Выполнение примера 5

**Пример 6.** Использование формы при решении задач: 1. В одномерном массиве нулевые элементы удалить, положительные элементы расставить по убыванию, отрицательные – по возрастанию. Получить зависимость затрат машинного времени от размера массива. 2. В матрице удалить строки с последними отрицательными элементами, а затем добавить строку из сумм элементов по столбцам.

Проверка работы с массивом (рисунок 6).

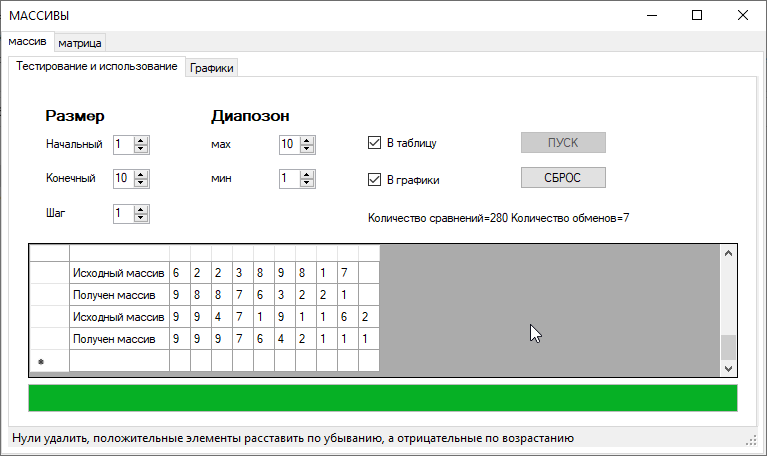


Рисунок 6 – Проверка работа с массивом

Проверка построения графиков (рисунок 7).

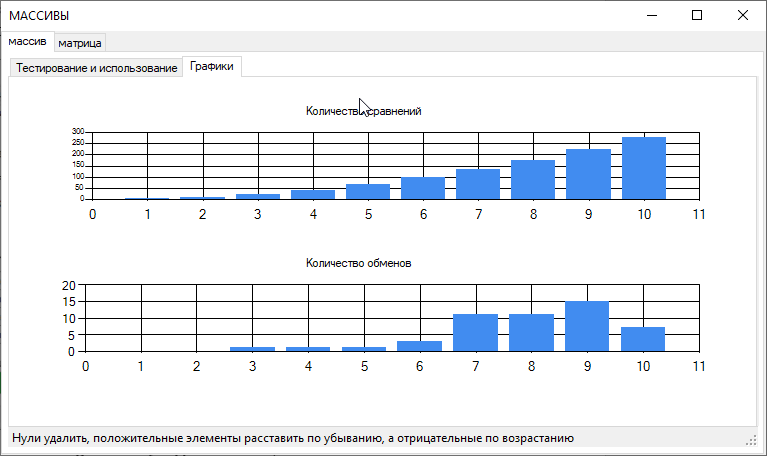


Рисунок 7 – Проверка построения графиков

Проверка работы с матрицами (рисунок 9).

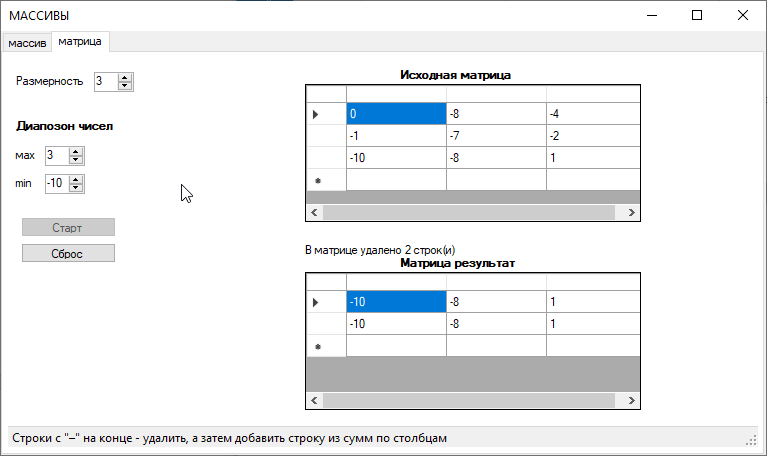


Рисунок 9 – Проверка работы с матрицами

**Задание 1.** Дан массив из координат х N точек на прямой. Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна, и саму эту сумму.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

namespace Задание\_1

{

class Program

{

static void printArr(int[] arr, string msg = "")

{

if(msg != "")

Console.WriteLine("{0}:",msg);

Console.Write("[");

foreach (var i in arr)

{

Console.Write("{0}, ", i);

}

Console.WriteLine("\b\b]");

}

static string printIt(int it, int mask, char spaser)

{

StringBuilder outIt = new StringBuilder(it.ToString());

while (outIt.Length != mask.ToString().Length)

{

outIt.Insert(0, spaser);

}

return outIt.ToString();

}

static void Main(string[] args)

{

int MAXVALUE = 100;

int MINVALUE = -100;

uint N = 10;

uint min\_i = 0;

int min\_sum = 0;

/\* Console.Write("Введите количество точек: ");

ushort N = ushort.Parse(Console.ReadLine()); // Количество точек\*/

int[] arr = new int[N]; // Массив точек

Random rand = new Random();

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

{

arr[i] = rand.Next(MINVALUE, MAXVALUE);

}

Array.Sort(arr);

printArr(arr, "Точки");

Console.WriteLine("\nСумма растояний для каждой точки:");

bool isInitSum = false;

min\_i = 0;

min\_sum = arr[0];

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

{

Console.Write("{0} ->\t", printIt(arr[i], MAXVALUE \* 10 \* 10, ' '));

int sum = 0;

foreach (var ii in arr)

{

int distanse = Math.Abs(ii - arr[i]);

sum += distanse;

Console.Write("{0}, ", printIt(distanse,MAXVALUE, ' '));

}

if(!isInitSum)

{

min\_sum = sum;

isInitSum = true;

}

if(sum < min\_sum && isInitSum)

{

min\_sum = sum;

min\_i = i;

}

Console.WriteLine("\b\b =\t {0}", sum);

}

Console.WriteLine("\nМинимальная сумма = {0}", min\_sum);

Console.WriteLine("Порядковый номер точки с минимальной суммой = {0}", min\_i + 1);

Console.WriteLine("Точка с минимальной суммой = {0}", arr[min\_i]);

Console.WriteLine("\n{{sum={0}, i={1}, point={2}}}\n", min\_sum, min\_i, arr[min\_i]);

Console.ReadKey(true);

}

}

}

Проверка работы программы (рисунок 10).

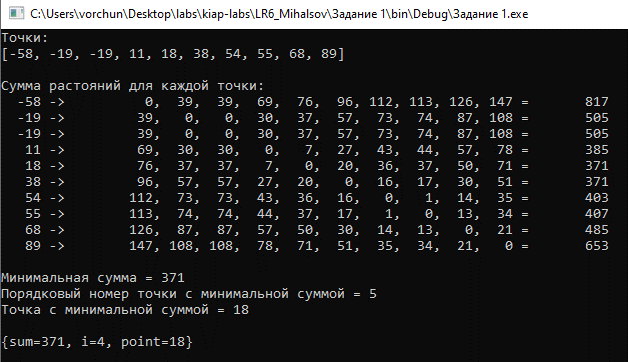


Рисунок 10 – Проверка работы программы

**Задание 2.** Характеристикой строки матрицы назовём сумму её отрицательных четных элементов. Расположить строки в соответствии с убыванием характеристик.

Листинг программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Задание\_2

{

class Program

{

static void Main(string[] args)

{

int N = 5;

int M = 5;

int MAXVALUE = 0;

int MINVALUE = -100;

int maxStringLong;

if(MAXVALUE.ToString().Length >= MINVALUE.ToString().Length)

{

maxStringLong = MAXVALUE.ToString().Length;

}

else

{

maxStringLong = MINVALUE.ToString().Length;

}

string maxString= new string('0', maxStringLong);

string FormatElem = maxString + ";" + "-" + maxString.Remove(0, 1) + ";" + maxString;

int[,] Matrix = new int[N, M];

int[] ArrOfPower = new int[N];

int[] SortArrOfPower = new int[M];

int[,] PowerSortMatrix = new int[N, M];

// Заполняем матрицу случайными значениями

Random rand = new Random();

for(int i = 0; i < N; ++i)

{

for (int j = 0; j < M; ++j)

{

Matrix[i, j] = rand.Next(MINVALUE, MAXVALUE);

}

}

// Считаем сумму по столбцам и записываем их массив

for(int i = 0; i < M; ++i)

{

int SumOfCol = 0;

for (int j = 0; j < N; ++j)

{

if(Matrix[i,j] < 0 && Matrix[i, j] % 2 != 0)

{

SumOfCol += Math.Abs(Matrix[i, j]);

}

}

ArrOfPower[i] = SumOfCol;

}

// Копируем массив с суммами в новый массив

Array.Copy(ArrOfPower, SortArrOfPower, ArrOfPower.Length);

// Сортируем значения в массиве

Array.Sort(SortArrOfPower);

// Записываем новую матрицу в таблицу

// Создаём переменную для столбцов матрицы

int ii = 0;

// Проходим по отсоритрованному массиву

foreach (var colum in SortArrOfPower)

{

// Ищем элемент в неотсорированном масиве (столбец)

int i = Array.IndexOf(ArrOfPower, colum);

// Перезаписываем значения в новую матрицу

for (int j = 0; j < N; ++j)

{

PowerSortMatrix[ii, j] = Matrix[i, j];

}

ii++;

}

// Выводим матрицы

for (int j = 0; j < M; j++)

{

for (int i = 0; i < N; i++)

{

Console.Write(String.Format(Matrix[i, j].ToString(FormatElem) + " "));

}

if(j == M % 2)

{

Console.Write(" -> ");

}

else

{

Console.Write(" ");

}

for (int i = 0; i < N; i++)

{

Console.Write(String.Format(PowerSortMatrix[i, j].ToString(FormatElem) + " "));

}

Console.WriteLine();

}

Console.WriteLine();

foreach (var i in ArrOfPower)

Console.Write(i.ToString(FormatElem) + " ");

Console.Write(" ");

foreach (var i in SortArrOfPower)

Console.Write(i.ToString(FormatElem) + " ");

Console.WriteLine();

double n = 9.3;

Console.ReadKey(true);

}

}

}

Проверка работы программы (рисунок 11).

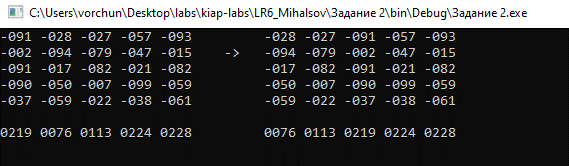


Рисунок 11 – Выполнение работы программы

**Вывод:** В результате выполнения лабораторной работы были получены практические навыки по работе с классом данных Array, его методами и свойствами.