Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

Объектно-ориентированные возможности языка C#

тема

Вариант 14

Преподаватель \_\_\_\_\_\_\_\_\_\_\_ А.А. Чикизов

подпись, дата инициалы, фамилия

Студент КИ15–16Б, 031510065 \_\_\_\_\_\_\_\_\_\_\_ Т.В. Радионов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2017

# Задание

Описать класс для работы с одномерным массивом:

- конструктор, заполняющий заданное количество элементов массива заданным значением;

- конструктор, заполняющий заданное в его параметре количество элементов массива значениями членов ряда Тейлора для функции cos x для заданного х.

- свойство, доступное только для чтения, для получения количества элементов массива, модуль которых больше 0,5;

- метод, вычисляющий сумму модулей элементов, расположенных после первого элемента, равного нулю

Вывод на экран выполнять только в методе Main класса-клиента. Программа должна адекватно реагировать на ошибки пользователя и различные варианты исходных данных. Все тестовые данные предъявить преподавателю.

# Исходный код

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

static void Main(string[] args)

{

int n = 0;

Console.Write("Введите размер массива: ");

try

{

n = Convert.ToInt32(Console.ReadLine());

}

catch (FormatException)

{

Console.Write("Введено не число");

Console.ReadKey();

return;

}

Array array = new Array(n);

Console.WriteLine("Размер массива: " + array.ArraySize);

Console.Write("Введите имя файла: ");

string fileName = Console.ReadLine();

array = new Array(n, fileName);

try

{

Console.WriteLine("Размер массива: " + array.ArraySize);

}

catch (NullReferenceException)

{

Console.Write("Ошибка имени файла");

Console.ReadKey();

return;

}

int result = array.Calculate();

Console.WriteLine("Результат произведения между минимальным и максимальным элементом: " + result.ToString());

Console.ReadKey();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace ConsoleApp1

{

class Array

{

int[] array; // Поле массив

// Св-во со значением размера массива

public int ArraySize

{

get

{

return array.Length;

}

}

public Array(int n)

{

array = new int[n];

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

{

array[i] = (i + 1) \* (i + 1);

}

}

public Array(int n, string fileName)

{

string path = fileName + ".txt";

FileStream file;

try

{

file = new FileStream(path, FileMode.Open, FileAccess.Read);

}

catch (FileNotFoundException)

{

return;

}

file = new FileStream(path, FileMode.Open, FileAccess.Read);

StreamReader reader = new StreamReader(file); // Потоковое чтение файла

while (!reader.EndOfStream)

{

string buf; // Буферная переменная для считываемой строки

buf = reader.ReadLine();

string[] bufArr = buf.Split(' '); // Создаем буферный массив с разбиением на эл-ты по пробелу

// Если переданное кол-во элементов больше, чем их есть в файле, то считываем все элементы файла

if (n > bufArr.Length)

{

n = bufArr.Length;

}

array = new int[n];

// Заполняем массив считанными данными

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

{

array[i] = Convert.ToInt32(bufArr[i]);

}

}

reader.Close();

}

public int Calculate()

{

int maxPos = 0;

int minPos = array.Length - 1;

int result = 1;

for (int i = 0; i < array.Length - 1; i++)

{

if (array[i + 1] >= array[maxPos])

{

maxPos = i + 1;

}

if (array[i + 1] <= array[minPos])

{

minPos = i + 1;

}

}

if (minPos > maxPos)

{

for (int j = maxPos + 1; j < minPos; j++)

{

result = result \* array[j];

}

}

else

{

for (int j = minPos + 1; j < maxPos; j++)

{

result = result \* array[j];

}

}

return result;

}

}

}

# Результы

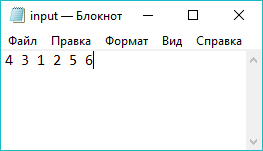
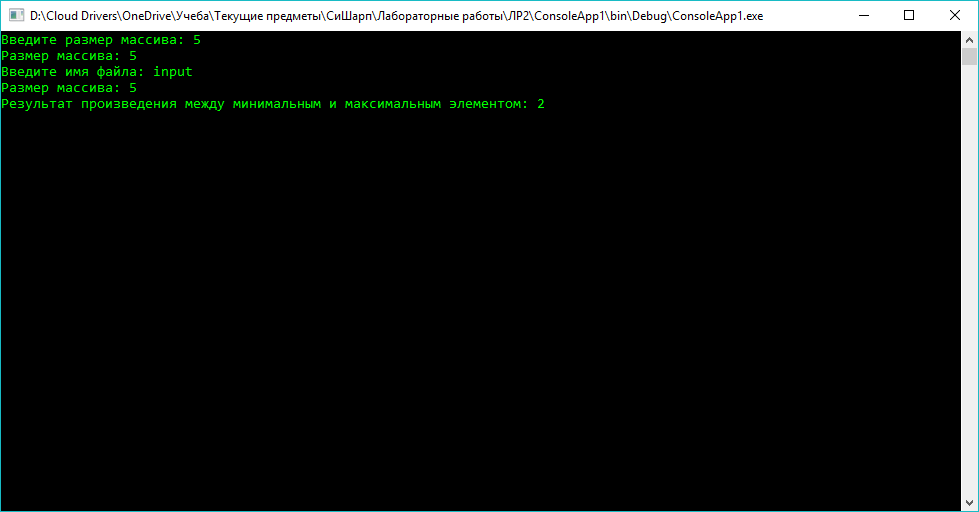


Рисунок 1 – Пример работы программы

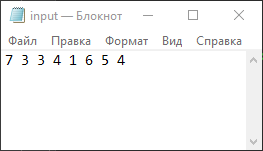
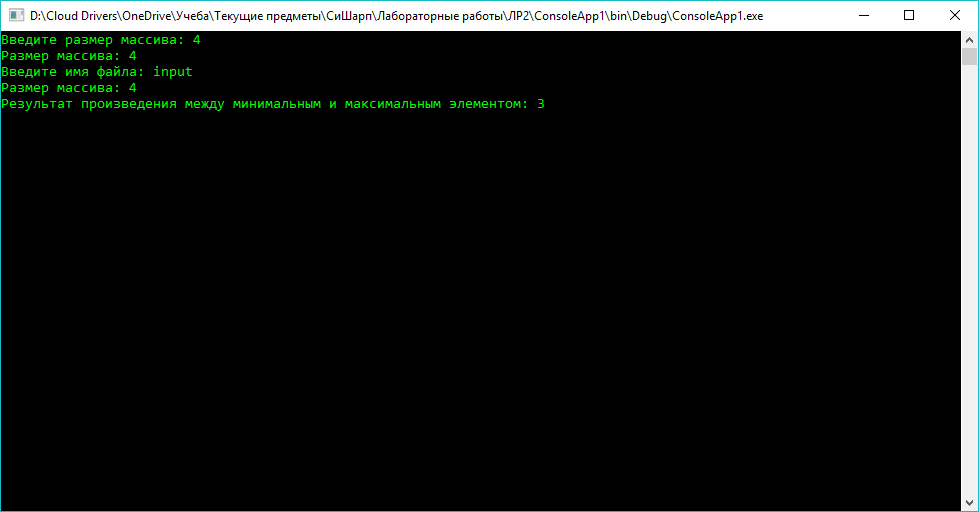


Рисунок 2 – Пример работы программы