Массивы и строки **НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Кафедра ИМиИКТ им. В.В. Дика |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные технологии и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | Очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по практической работе №4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Массивы и строки | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | **Разработка модулей программного обеспечения для компьютерных систем** |
|  | | |  | (наименование дисциплины) |

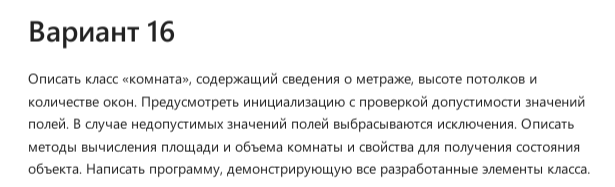
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Мифтахов Михаил Серикжанович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | Группа – ДКИП-206прог |  |  |
|  |  |  |  |  |

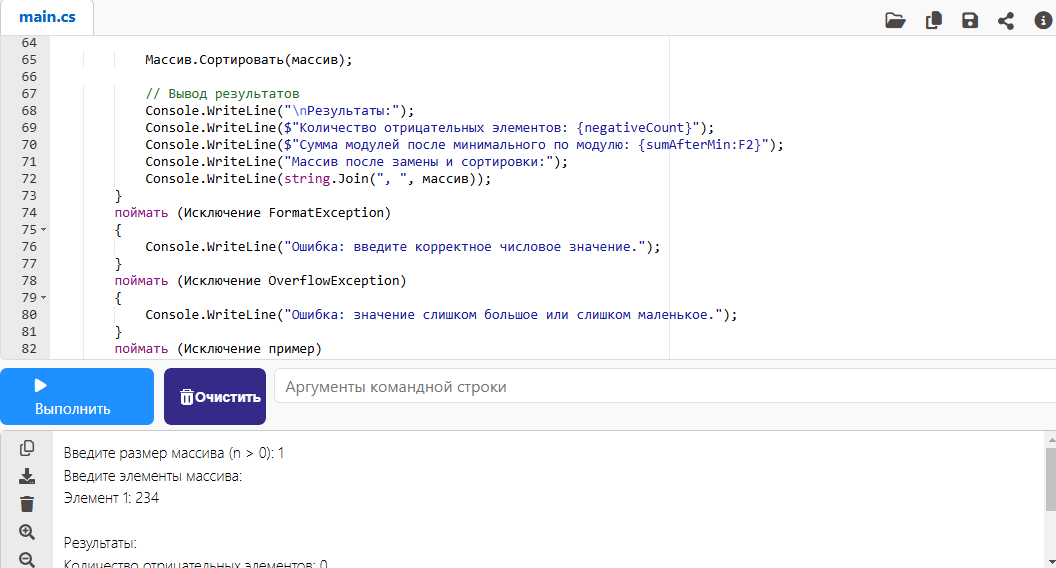
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | **Сибирев Иван Валерьевич** |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г.**

Лабораторная работа №4 «Массивы и строки».

ЛАБ. №5





using System;

class Program

{

static void Main()

{

try

{

// Ввод размера массива

Console.Write("Введите размер массива (n > 0): ");

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

if (n <= 0)

{

Console.WriteLine("Ошибка: Размер массива должен быть положительным.");

return;

}

// Инициализация и заполнение массива

double[] array = new double[n];

Console.WriteLine("Введите элементы массива:");

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

{

Console.Write($"Элемент {i + 1}: ");

array[i] = Convert.ToDouble(Console.ReadLine());

}

// 1. Количество отрицательных элементов

int negativeCount = 0;

foreach (double num in array)

{

if (num < 0)

negativeCount++;

}

// 2. Сумма модулей элементов после минимального по модулю

int minIndex = 0;

double minAbsValue = Math.Abs(array[0]);

for (int i = 1; i < n; i++)

{

double currentAbs = Math.Abs(array[i]);

if (currentAbs < minAbsValue)

{

minAbsValue = currentAbs;

minIndex = i;

}

}

double sumAfterMin = 0;

for (int i = minIndex + 1; i < n; i++)

{

sumAfterMin += Math.Abs(array[i]);

}

// 3. Замена отрицательных элементов на квадраты и сортировка

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

{

if (array[i] < 0)

{

array[i] = array[i] \* array[i];

}

}

Array.Sort(array);

// Вывод результатов

Console.WriteLine("\nРезультаты:");

Console.WriteLine($"Количество отрицательных элементов: {negativeCount}");

Console.WriteLine($"Сумма модулей после минимального по модулю: {sumAfterMin:F2}");

Console.WriteLine("Массив после замены и сортировки:");

Console.WriteLine(string.Join(", ", array));

}

catch (FormatException)

{

Console.WriteLine("Ошибка: Введите корректное числовое значение.");

}

catch (OverflowException)

{

Console.WriteLine("Ошибка: Значение слишком большое или маленькое.");

}

catch (Exception ex)

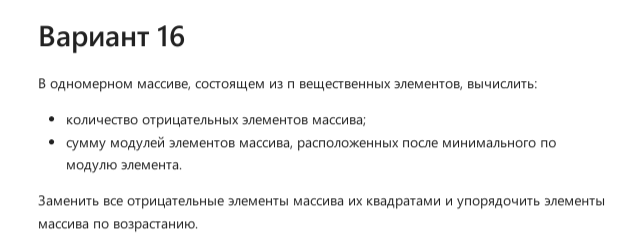
{

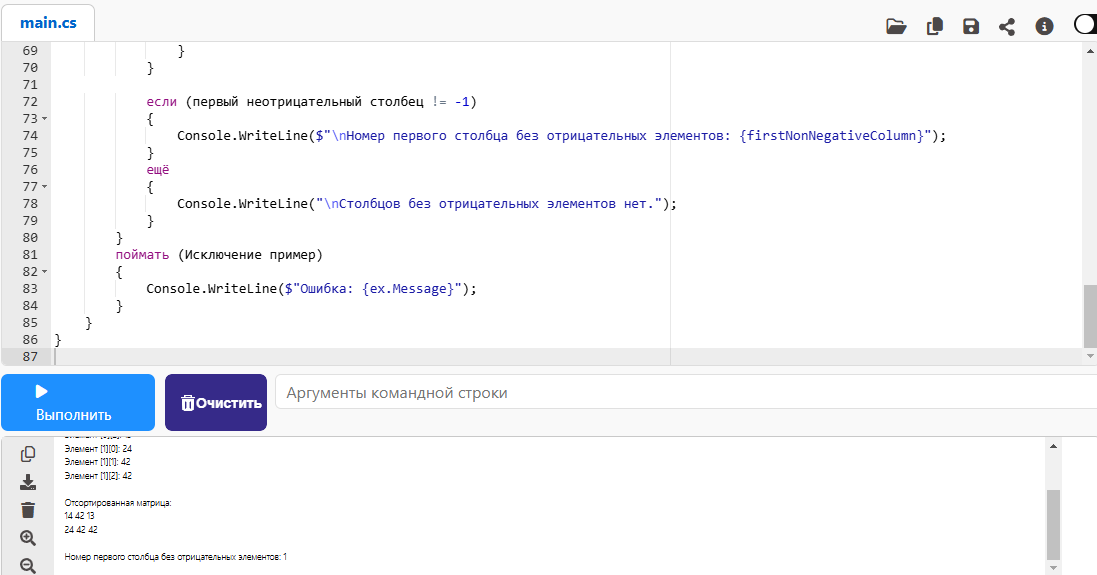
Console.WriteLine($"Произошла ошибка: {ex.Message}");

}

}

}

ЛАБ. №6



using System;

using System.Linq;

class Program

{

static void Main()

{

try

{

// Ввод размеров матрицы

Console.Write("Введите количество строк: ");

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

Console.Write("Введите количество столбцов: ");

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

// Инициализация матрицы

int[][] matrix = new int[rows][];

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

{

matrix[i] = new int[cols];

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

{

Console.Write($"Элемент [{i}][{j}]: ");

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

}

}

// Функция для подсчета максимального количества одинаковых элементов в строке

Func<int[], int> countMaxDuplicates = row =>

{

var counts = new System.Collections.Generic.Dictionary<int, int>();

foreach (int num in row)

{

if (counts.ContainsKey(num))

counts[num]++;

else

counts[num] = 1;

}

return counts.Values.Max();

};

// Сортировка строк по возрастанию количества одинаковых элементов

Array.Sort(matrix, (row1, row2) => countMaxDuplicates(row1).CompareTo(countMaxDuplicates(row2)));

// Вывод отсортированной матрицы

Console.WriteLine("\nОтсортированная матрица:");

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

{

Console.WriteLine(string.Join(" ", matrix[i]));

}

// Поиск первого столбца без отрицательных элементов

int firstNonNegativeColumn = -1;

for (int col = 0; col < cols; col++)

{

bool hasNegative = false;

for (int row = 0; row < rows; row++)

{

if (matrix[row][col] < 0)

{

hasNegative = true;

break;

}

}

if (!hasNegative)

{

firstNonNegativeColumn = col + 1; // нумерация с 1

break;

}

}

if (firstNonNegativeColumn != -1)

{

Console.WriteLine($"\nНомер первого столбца без отрицательных элементов: {firstNonNegativeColumn}");

}

else

{

Console.WriteLine("\nСтолбцов без отрицательных элементов нет.");

}

}

catch (Exception ex)

{

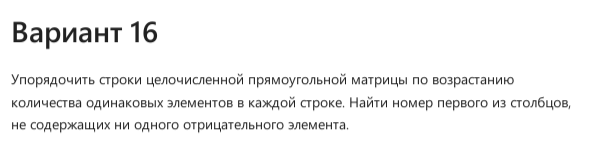
Console.WriteLine($"Ошибка: {ex.Message}");

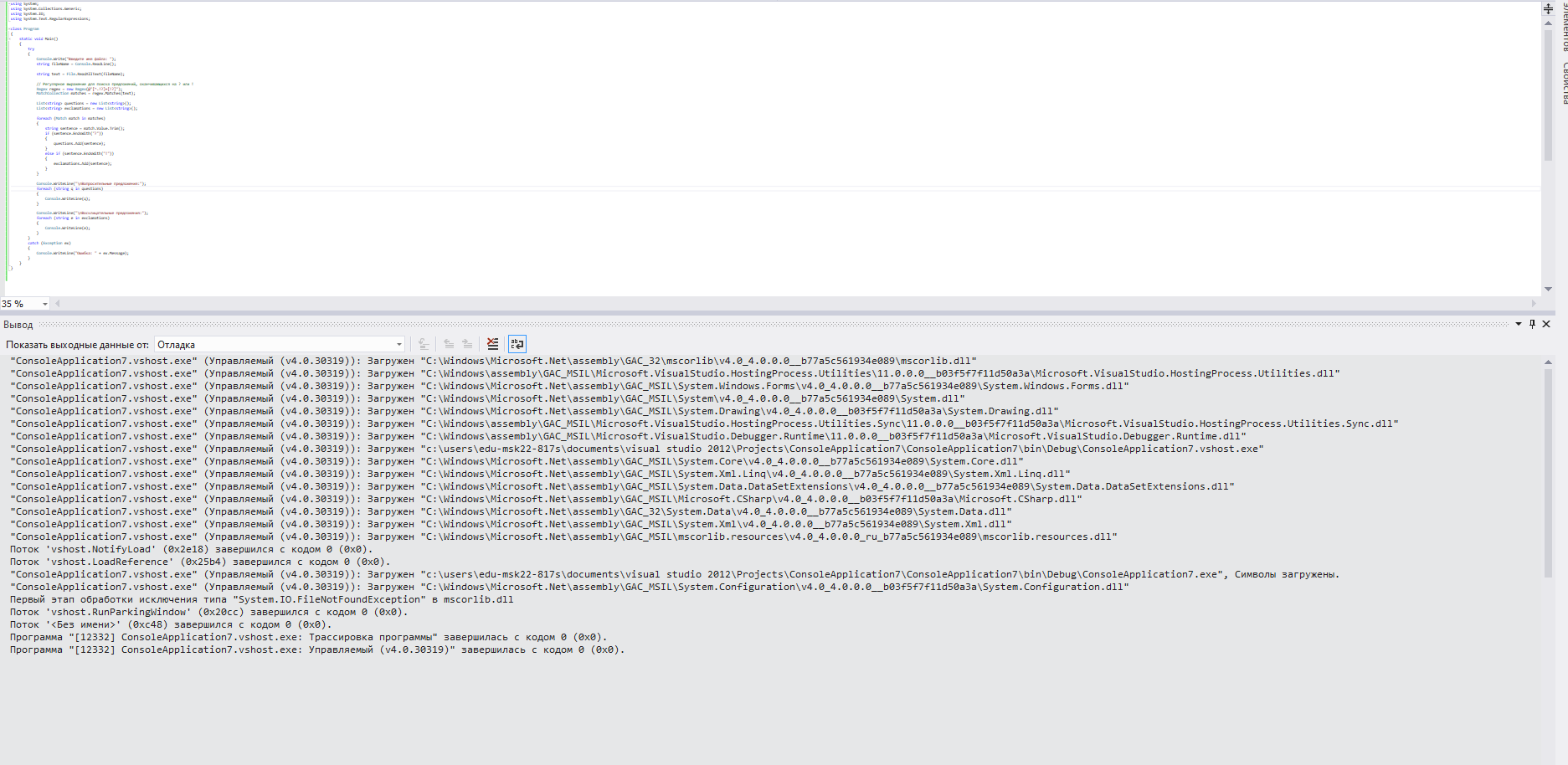
}

}

}

ЛАБ. №7





using System;

using System.Collections.Generic;

using System.IO;

using System.Text.RegularExpressions;

class Program

{

static void Main()

{

try

{

Console.Write("Введите имя файла: ");

string fileName = Console.ReadLine();

string text = File.ReadAllText(fileName);

// Регулярное выражение для поиска предложений, оканчивающихся на ? или !

Regex regex = new Regex(@"[^.!?]+[!?]");

MatchCollection matches = regex.Matches(text);

List<string> questions = new List<string>();

List<string> exclamations = new List<string>();

foreach (Match match in matches)

{

string sentence = match.Value.Trim();

if (sentence.EndsWith("?"))

{

questions.Add(sentence);

}

else if (sentence.EndsWith("!"))

{

exclamations.Add(sentence);

}

}

Console.WriteLine("\nВопросительные предложения:");

foreach (string q in questions)

{

Console.WriteLine(q);

}

Console.WriteLine("\nВосклицательные предложения:");

foreach (string e in exclamations)

{

Console.WriteLine(e);

}

}

catch (Exception ex)

{

Console.WriteLine("Ошибка: " + ex.Message);

}

}

}