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

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Программирование |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | Очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по практической работе №1**

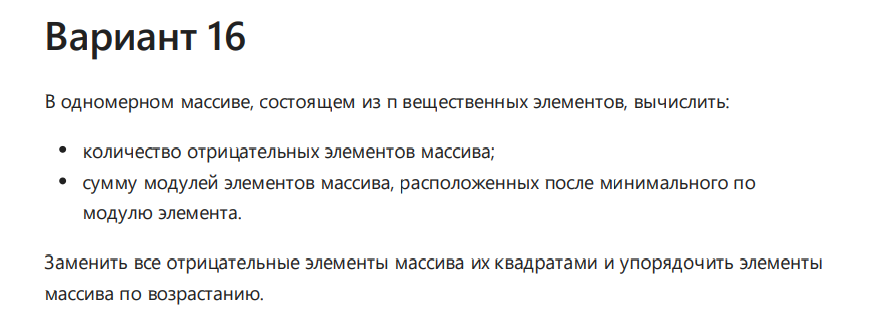
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Использование операторов ветвления | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Разработка программных модулем |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Стрельцов Егор Михайлович |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | VДКИП-111прог |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев И. В. |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г.**

Лабораторная работа 5. Одномерные массивы. Вариант 16



Листинг кода:

using System;

using System.Linq;

using static System.Runtime.InteropServices.JavaScript.JSType;

class Program

{

static void Main()

{

Console.Write("Введите количество элементов массива:");

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

double[] array = new double[n];

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

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

{

Console.Write($"Элемент массива {i + 1}: ");

array[i] = double.Parse(Console.ReadLine());

}

int NegativeCount = array.Count(x => x < 0);

Console.WriteLine($"Колличество отрицательных элементов: {NegativeCount}");

double minAbs = array.Min(x => Math.Abs(x));

int minAbsIndex = Array.FindIndex(array, x => Math.Abs(x) == minAbs);

double Sum = 0;

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

{

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

}

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

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

{

if (array[i] < 0)

{

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

}

}

Array.Sort(array);

Console.WriteLine("Преобразованный массив: ");

foreach (var item in array)

{

Console.Write(item + " ");

}

}

}

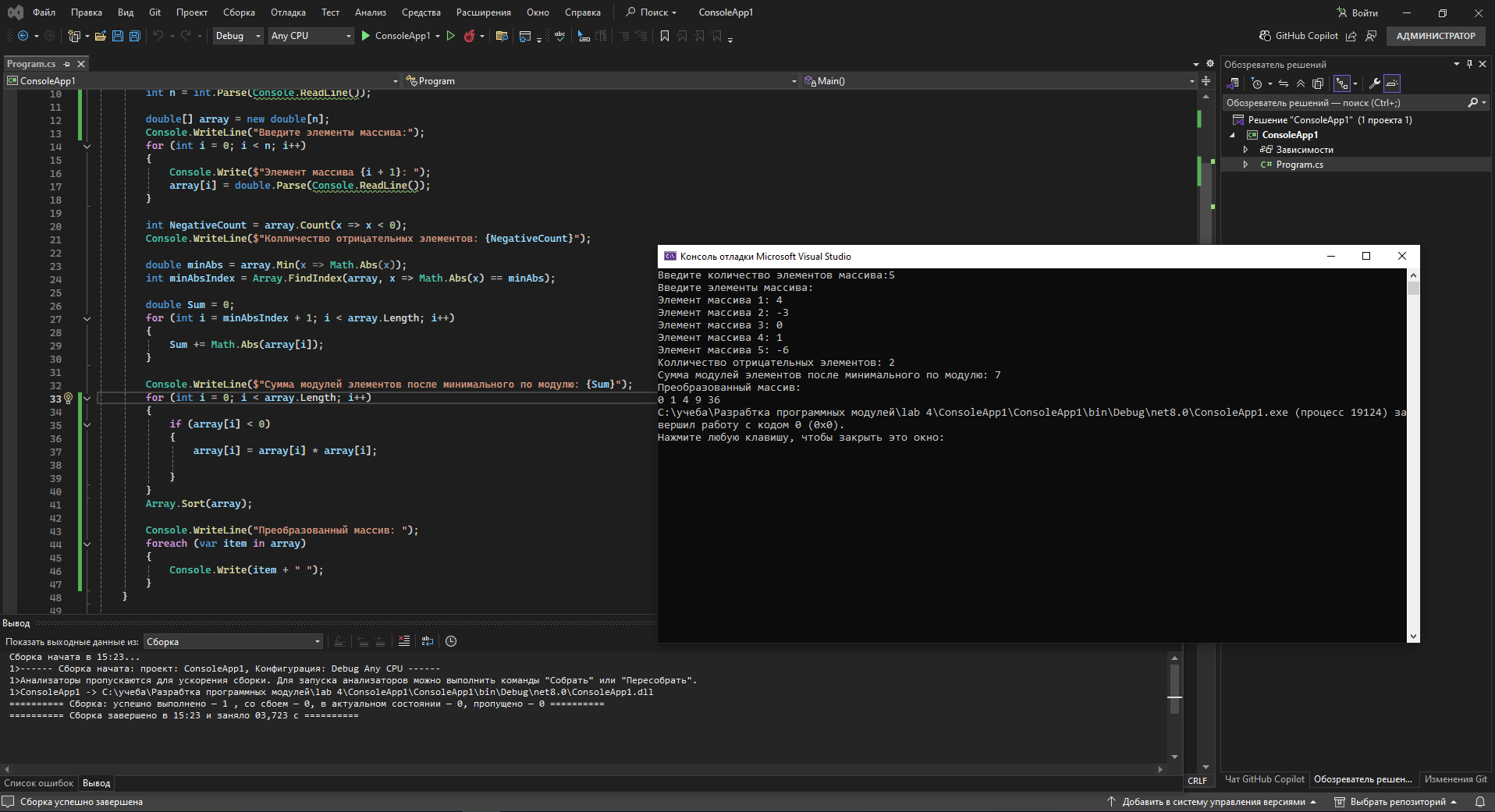
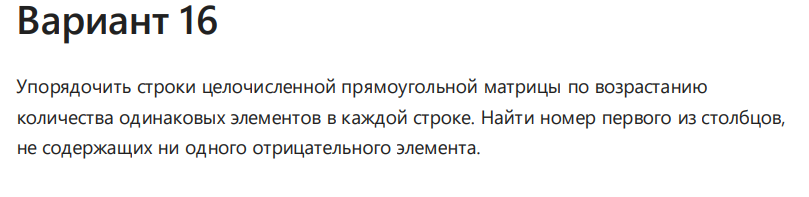


Рис. 1 Выполнение программы

Лабораторная работа 6. Двумерные массивы.



Листинг кода:

using System;

using System.Linq;

class Program

{

static void Main()

{

// Пример матрицы

int[,] matrix = {

{ 1, 2, 3, 4 },

{ 5, 5, 6, 6 },

{ 7, 7, 7, 8 },

{ 9, 9, 9, 9 }

};

Console.WriteLine("Исходная матрица:");

PrintMatrix(matrix);

// 1. Упорядочивание строк по количеству одинаковых элементов

var orderedMatrix = OrderRowsByDuplicateCount(matrix);

Console.WriteLine("\nМатрица после сортировки строк:");

PrintMatrix(orderedMatrix);

// 2. Поиск первого столбца без отрицательных элементов

int columnIndex = FindFirstColumnWithoutNegatives(matrix);

if (columnIndex >= 0)

Console.WriteLine($"\nПервый столбец без отрицательных элементов: {columnIndex}");

else

Console.WriteLine("\nНет столбцов без отрицательных элементов");

}

// Метод для упорядочивания строк матрицы по количеству одинаковых элементов

static int[,] OrderRowsByDuplicateCount(int[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

// Преобразуем матрицу в массив строк для удобства сортировки

int[][] matrixAsRows = new int[rows][];

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

{

matrixAsRows[i] = new int[cols];

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

{

matrixAsRows[i][j] = matrix[i, j];

}

}

// Сортируем строки по количеству одинаковых элементов (используем LINQ)

var sortedRows = matrixAsRows

.OrderBy(row => row

.GroupBy(x => x)

.Sum(g => g.Count() - 1))

.ToArray();

// Возвращаем результат в виде двумерного массива

int[,] result = new int[rows, cols];

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

{

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

{

result[i, j] = sortedRows[i][j];

}

}

return result;

}

// Метод для поиска первого столбца без отрицательных элементов

static int FindFirstColumnWithoutNegatives(int[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

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

{

bool hasNegative = false;

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

{

if (matrix[i, j] < 0)

{

hasNegative = true;

break;

}

}

if (!hasNegative)

return j;

}

return -1; // если не найден

}

// Вспомогательный метод для вывода матрицы

static void PrintMatrix(int[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

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

{

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

{

Console.Write($"{matrix[i, j],4}");

}

Console.WriteLine();

}

}

}

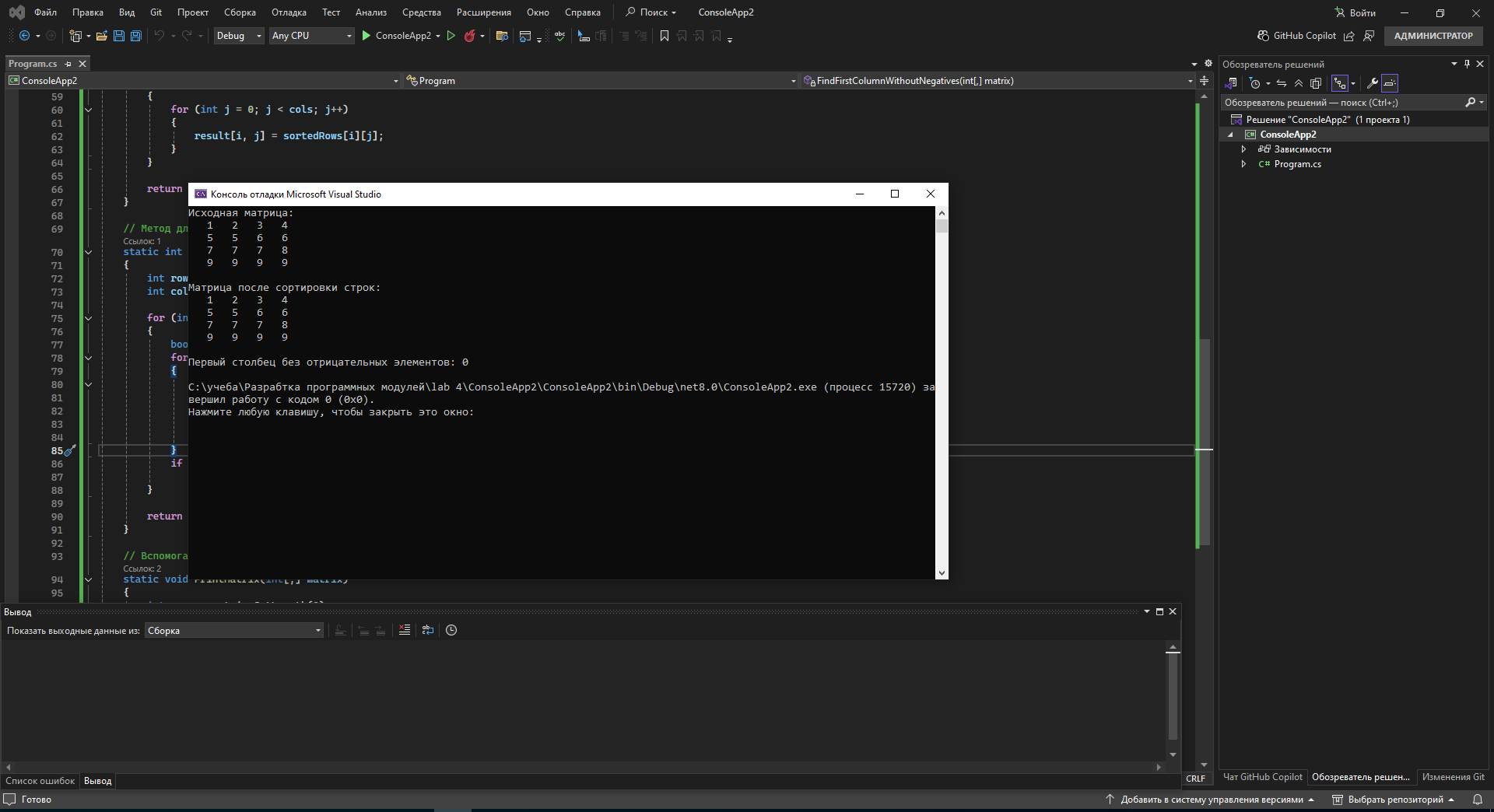
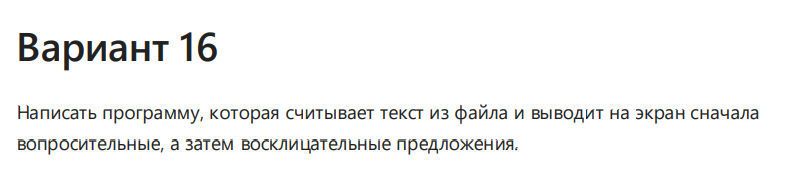


Рис. 2. Выполнение программы

Лабораторная работа 7. Строки. Вариант 16.



Листинг кода:

using System;

using System.IO;

using System.Text;

class Program

{

static void Main()

{

string filePath = @"C:\test\_file.txt"; // Укажите путь к вашему файлу

try

{

// Чтение всего текста из файла

string text = File.ReadAllText(filePath, Encoding.UTF8);

// Разделение текста на предложения

string[] sentences = text.Split(new[] { '.', '?', '!' }, StringSplitOptions.RemoveEmptyEntries);

char[] delimiters = text.Where(c => c == '?' || c == '!').ToArray();

// Собираем предложения с их разделителями

var sentenceList = new System.Collections.Generic.List<(string sentence, char delimiter)>();

for (int i = 0; i < Math.Min(sentences.Length, delimiters.Length); i++)

{

sentenceList.Add((sentences[i].Trim(), delimiters[i]));

}

// Выбираем вопросительные предложения

var questions = sentenceList.Where(x => x.delimiter == '?')

.Select(x => x.sentence + x.delimiter);

// Выбираем восклицательные предложения

var exclamations = sentenceList.Where(x => x.delimiter == '!')

.Select(x => x.sentence + x.delimiter);

// Выводим результат

Console.WriteLine("Вопросительные предложения:");

foreach (var q in questions)

{

Console.WriteLine(q);

}

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

foreach (var e in exclamations)

{

Console.WriteLine(e);

}

}

catch (FileNotFoundException)

{

Console.WriteLine("Файл не найден!");

}

catch (Exception ex)

{

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

}

}

}

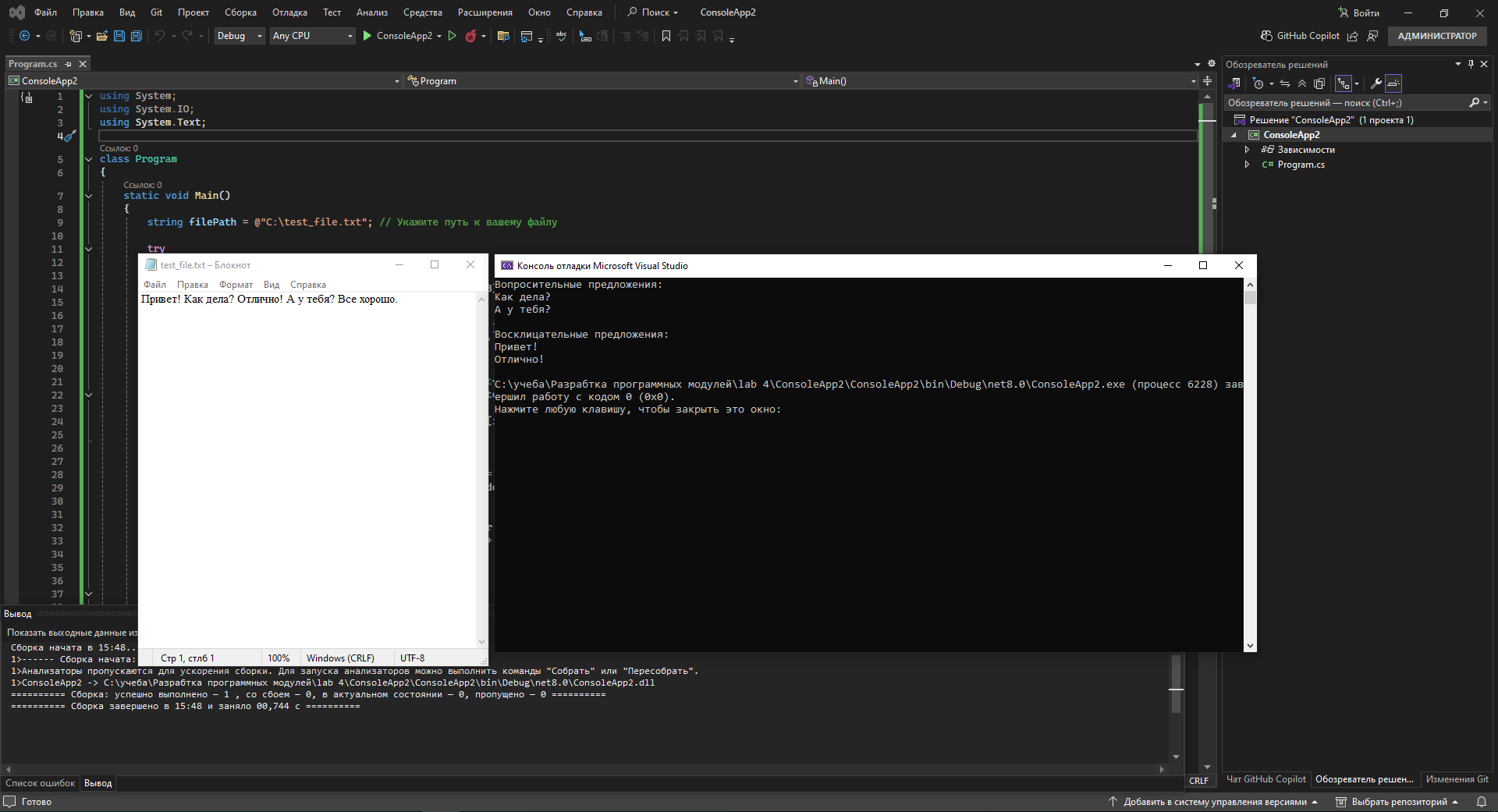


Рис. 3. Выполнение программы