**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**КАФЕДРА САПР**

**ОТЧЕТ**

**по лабораторной работе №5**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: Создание и использование массивов**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 9301 |  | Синицкая В. А. |
| Преподаватель |  | Новакова Н.Е. |

Санкт-Петербург

2021

Оглавление

[1 Цель работы 3](#_Toc65497729)

[2 Анализ задачи 3](#_Toc65497730)

[3 Формальная постановка задачи 3](#_Toc65497731)

[3.1 Исходные данные 3](#_Toc65497732)

[3.2 Результирующие (выходные) данные 3](#_Toc65497733)

[4 Используемые классы и методы 4](#_Toc65497734)

[5 Текст программы 5](#_Toc65497735)

[5.1 Текст программы 1 5](#_Toc65497736)

[5.2 Текст программы 2 8](#_Toc65497737)

[6 Контрольный пример 9](#_Toc65497738)

[6.1 Контрольный пример для 1 программы 9](#_Toc65497739)

[6.2 Контрольный пример для 2 программы 10](#_Toc65497740)

[7 Полученные результаты 11](#_Toc65497741)

[8 Выводы 11](#_Toc65497742)

[9 Список использованных источников 11](#_Toc65497743)

# Цель работы

Получение навыков работы с чтением из файла и с консоли, практика использования одномерных и двумерных массивов. Работа с командной строкой: вызов программы и передача аргументов, работа с методами, их вызовом и спецификаторами передаваемых параметров.

# Анализ задачи

Программа:

Написать программу, запускаемую из командной строки, в качестве аргумента передать имя текстового файла, выполнить анализ текста: подсчитать количество символов, гласных и согласных букв, количество строк.

Программа:

Написать программу, считывающую с консоли две матрицы размера 2x2, сохранить результат произведения первой на вторую и вывести пользователю в консоль.

# Формальная постановка задачи

## Исходные данные

Программа:

Имя текстового файла, как параметр вызова программы из командной строки, текст, содержащийся внутри него.

Программа:

2 пары по 4 целых числа, записанных в виде матриц.

## Результирующие (выходные) данные

Программа:

Текст файла, количество символов в текстовом файле, количество гласных и согласных букв, количество строк, либо сообщение об ошибке и о местоположении ее возникновения.

Программа:

2 пары по 4 целых числа, записанных в виде матриц.

# Используемые классы и методы

В программах используются методы Console.WriteLine() и Console.ReadLine() — методы класса System.Console. System.Console — это класс для работы с консольным окном, определенный разработчиками стандартных библиотек для языка C#.

Метод Console.ReadLine() сохраняет введенную пользователем строку в заданную переменную.

Метод Console.WriteLine() служит для отображения в консольном окне пользователя строк и других данных, переданных в данный метод в качестве параметров.

Программа:

Содержит единственный класс FileDetails, в котором два метода: Summarize и Main. Метод Main в качестве параметра принимает массив строк args, в args[0] содержится название текстового файла, если он не был передан, вызывается исключение. Исключение так же вызывается, если файл не был найден, либо пуст. С помощью метода FileStream создается поток stream для чтения из указанного файла. Затем, с помощью метода StreamReader создается переменная для чтения, связанная с указанным ранее потоком. Используя stream.Length, записывается длина файла, которая совпадает с искомым количеством символов. Создается массив символов того же размера, в него копируется весть текст посимвольно, затем этот массив в качестве параметра вместе с параметрами типа out передается в метод Summarize, который, используя цикл foreach по символам массива и метод IndexOf, подсчитывает количество строк, гласных и согласных букв. Далее метод Main выводит необходимую информацию в консоль.

Программа:

Содержит единственный класс MatrixMultiply, в котором один метод — Main. В этом методе создается три массива, два из них считываются с консоли, в третий записывается результат перемножения, и далее выводится в консоль, используя два цикла for, один из которых вложен в другой, во внутреннем используется метод Console.Write(), выводящий переменную без перехода на новую строку, во внешнем цикле используется уже WriteLine, так что итоговый вывод представляется пользователю в виде матрицы. В методе используется метод string.Split(), по заданному разделителю возвращающий массив строк из одной строки, чтобы корректно считать введенные пользователем матрицы.

# Текст программы

## Текст программы 1

1. using System;
2. using System.IO; // FileStream, FileReader
4. namespace OOP5LabaFileDetails
5. {
6. class FileDetails
7. {
8. static void Summarize(char[] char\_array, out long vowels, out long consonents, out long line\_breaks)
9. {
10. vowels = consonents = line\_breaks = 0;
11. foreach (char symbol in char\_array)
12. {
13. if ("AEIOUaeiou".IndexOf(symbol) != -1)
14. // symbol is a vowel
15. vowels++;
16. else if ("BCDFGHJKLMNPQRSTVWXYZbcdfghjklmnpqrstvwxyz".IndexOf(symbol) != -1)
17. // symbol is a consonent
18. consonents++;
19. else if (symbol == '\n') // symbol is a line break
20. line\_breaks++;
21. }
22. }
23. static void Main(string[] args)
24. {
25. //Console.WriteLine(args.Length);
26. //foreach (string string\_arg in args)
27. // Console.WriteLine(string\_arg);
28. try
29. {
30. try
31. {
32. if (args.Length < 1)
33. throw new IndexOutOfRangeException("start argument is missing");
34. string fileName = args[0]; // we sholud start program from cmd using fileName like parameter
35. FileStream stream = new FileStream(fileName, FileMode.Open);
36. StreamReader reader = new StreamReader(stream);
37. long length\_of\_file = stream.Length; // length of fileName file
38. if (!File.Exists(fileName)) // file doesn't exist
39. throw new FileNotFoundException("file doesn't exist");
40. if (length\_of\_file <= 0)
41. throw new ArgumentOutOfRangeException("file is empty");
42. char[] contents = new char[length\_of\_file]; // char massive, copy of fileName file
43. //Console.WriteLine("length = {0}", contents.Length);
44. for (int i = 0; i < length\_of\_file; i++)
45. contents[i] = (char)(reader.Read());
46. foreach (char symbol in contents)
47. Console.Write(symbol);
48. reader.Close();
49. long vowels\_amount, consonents\_amount, line\_breaks\_amount;
50. Summarize(contents, out vowels\_amount, out consonents\_amount, out line\_breaks\_amount);
51. Console.WriteLine("\nAmount of symbols in the file: {0}", contents.Length);
52. Console.WriteLine("Amount of vowels: {0}", vowels\_amount);
53. Console.WriteLine("Amount of consonents: {0}", consonents\_amount);
54. Console.WriteLine("Amount of lines: {0}", line\_breaks\_amount + 1); // last line doesn't have a line break
56. }
57. catch (IndexOutOfRangeException startArgument)
58. {
59. Console.WriteLine("Error: {0}", startArgument.Message);
60. Console.WriteLine("Stack: {0}", startArgument.StackTrace);
61. }
62. catch (ArgumentOutOfRangeException argument)
63. {
64. Console.WriteLine("Error: {0}", argument.Message);
65. Console.WriteLine("Stack: {0}", argument.StackTrace);
66. }
67. catch (FileNotFoundException noFile)
68. {
69. Console.WriteLine("Error: {0}", noFile.Message);
70. Console.WriteLine("Stack: {0}", noFile.StackTrace);
71. }
72. }
73. catch (Exception some\_exception)
74. {
75. Console.WriteLine("Error: {0}", some\_exception.Message);
76. Console.WriteLine("Stack: {0}", some\_exception.StackTrace);
77. }
78. }
79. }
80. }

## Текст программы 2

1. using System;
3. namespace OOP5LabaMatrix
4. {
5. class MatrixMultiply
6. {
7. static void Main(string[] args)
8. {
9. int[,] a = new int[2, 2];
10. int[,] b = new int[2, 2];
11. int[,] result = new int[2, 2] { { 0, 0 }, { 0, 0} };
13. Console.WriteLine("Enter first matrix 2x2");
15. for (int i = 0; i < 2; i++)
16. {
17. string[] data = Console.ReadLine().Split(new char[] { ' ' });
18. for (int j = 0; j < 2; j++)
19. {
20. a[i, j] = int.Parse(data[j]);
21. }
22. }
24. Console.WriteLine("Enter second matrix 2x2");
26. for (int i = 0; i < 2; i++)
27. {
28. string[] data = Console.ReadLine().Split(new char[] { ' ' });
29. for (int j = 0; j < 2; j++)
30. {
31. b[i, j] = int.Parse(data[j]);
32. }
33. }
35. for (int i = 0; i < 2; i++)
36. {
37. for (int j = 0; j < 2; j++)
38. {
39. for (int k = 0; k < 2; k++)
40. result[i, j] += a[i, k] \* b[k, j];
41. }
42. }
44. Console.WriteLine("the result of multiplication first on second:");
45. for (int i = 0; i < 2; i++)
46. {
47. for (int j = 0; j < 2; j++)
48. {
49. Console.Write(" {0}", result[i,j]);
50. }
51. Console.WriteLine();
52. }
53. }
54. }
55. }

# Контрольный пример

## Контрольный пример для 1 программы

На Рис. 6.1.1 показан вызов программы 1 из командной строки и результат работы программы: вывод текста файла и его анализа.

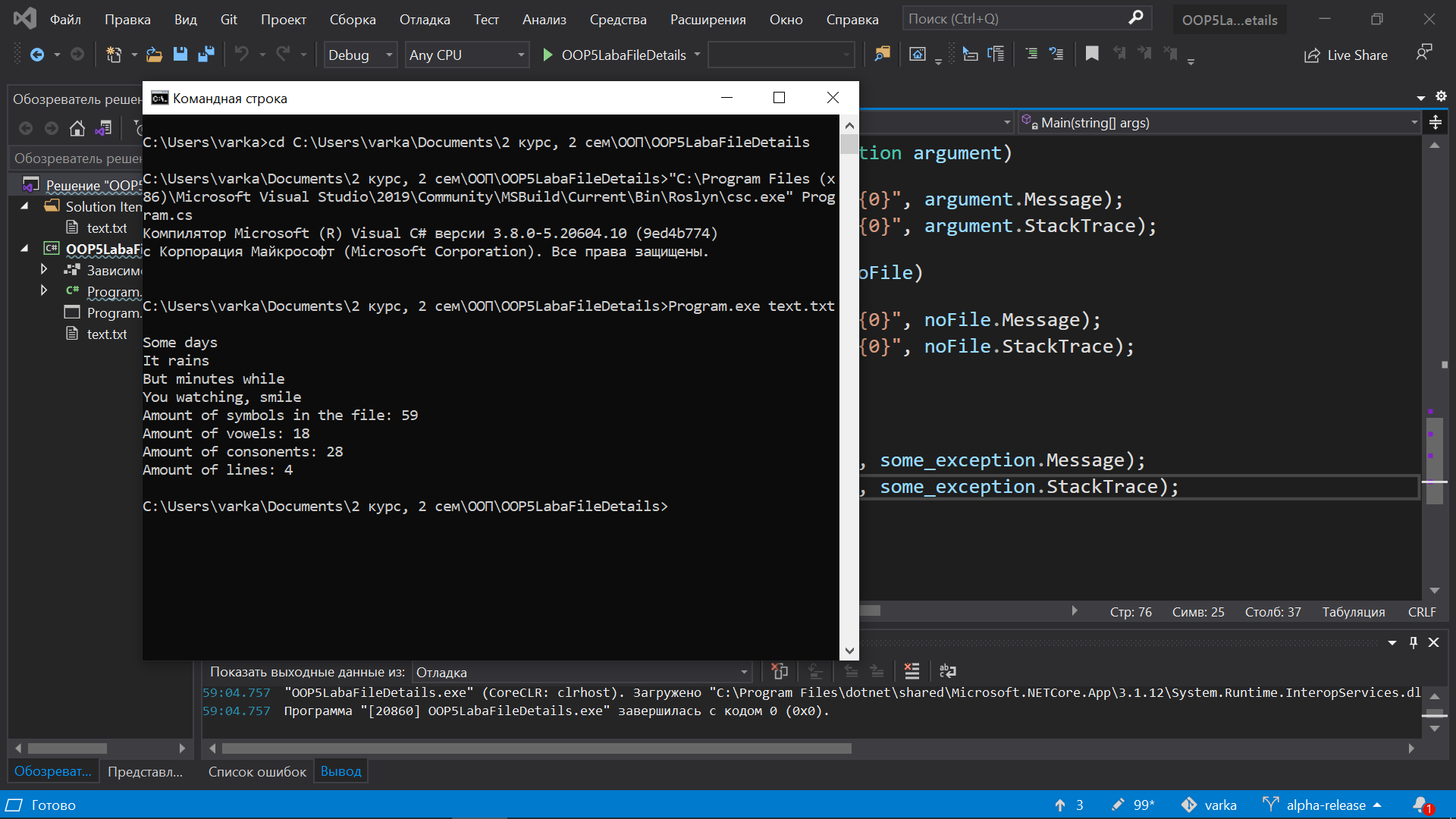


Рис. 6.1.1 — Вызов программы 1 из командной строки

На Рис. 6.1.2 показана обработка исключения при вызове программы без указания параметра — названия текстового файла.

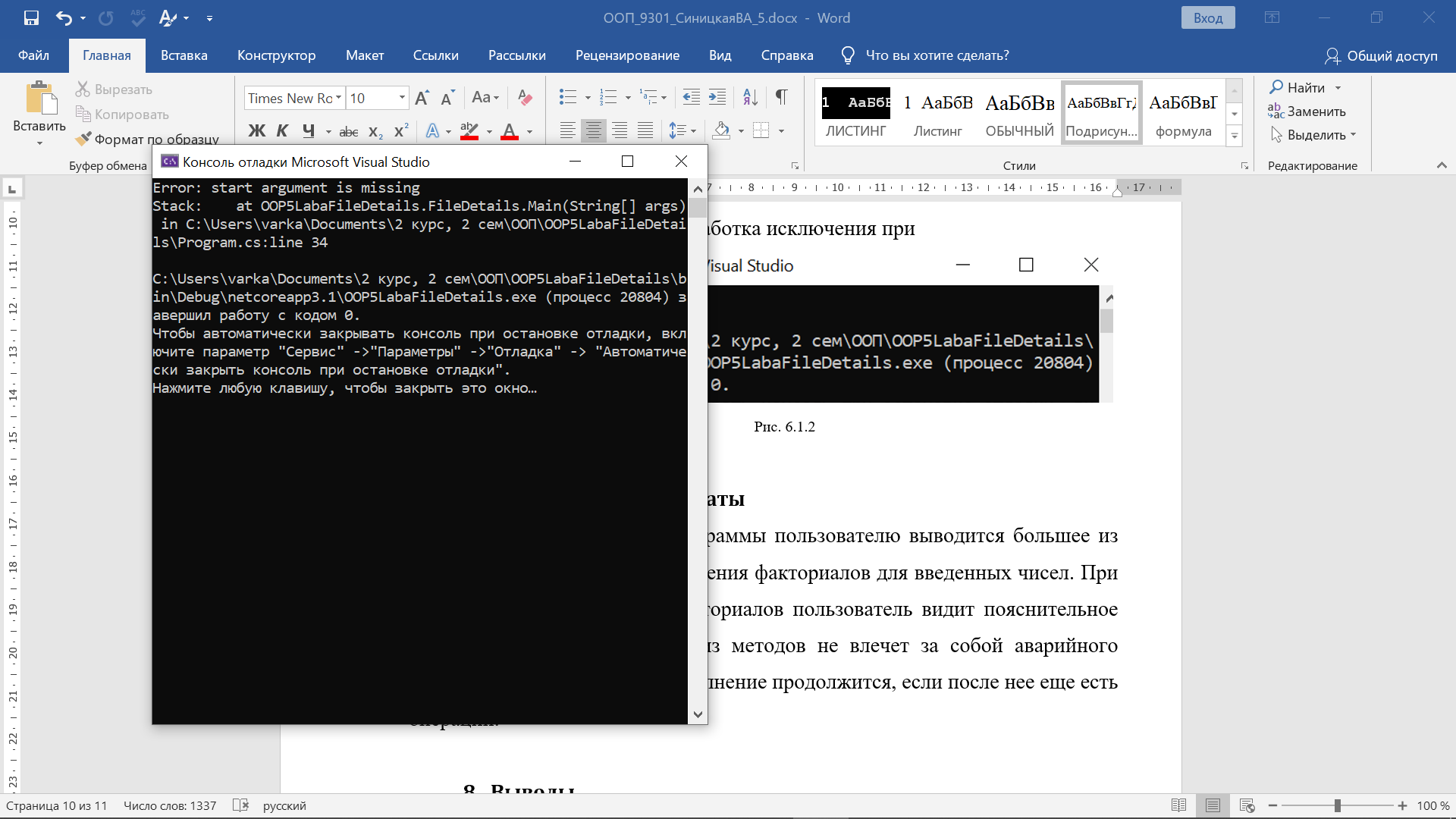


Рис. 6.1.2 — Обработка исключения программой 1

## Контрольный пример для 2 программы

На Рис. 6.2.1 показана работа программы 2.

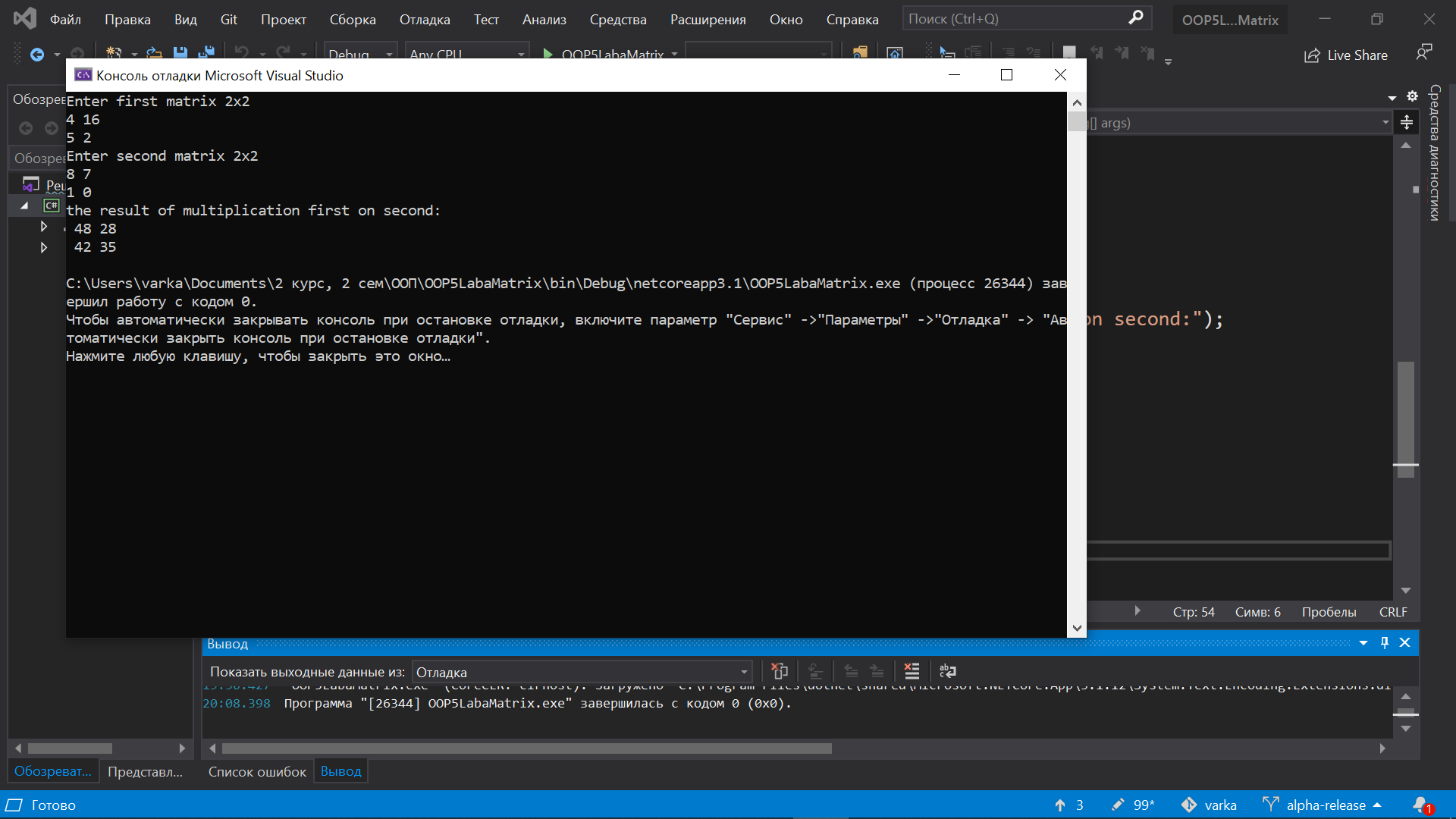


Рис. 6.2.1 — перемножение двух матриц программой 2.

# Полученные результаты

По результатам работы программы 1 был выведен текст указанного пользователем файла, количество символов в этом файле, количество гласных и согласных букв, количество строк.

По результатам работы программы 2 в консоль была выведена матрица, равная произведению двух данных.

# Выводы

В ходе выполнения данной лабораторной работы:

Были получены навыки использования одномерных и двумерных массивов.

Были изучены методы чтения из файла, способы преобразования полученных данных с консоли.

Практиковались навыки обращения со ссылочными параметрами при создании и вызове метода.

Были получены знания о работе с командной строкой.

# Список использованных источников

Руководство по языку C#: URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения: 01.03.2021)

Материалы учебного курса по ООП: URL: <https://vec.etu.ru/moodle/course/view.php?id=5512> (дата обращения: 01.03.2021)