**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**КАФЕДРА САПР**

**ОТЧЕТ**

**по лабораторной работе №1**

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

**Тема: Обзор языка C#**

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

Санкт-Петербург

2021

Оглавление

[1 Цель работы 3](#_Toc64299181)

[2 Анализ задачи 3](#_Toc64299182)

[3 Формальная постановка задачи 3](#_Toc64299183)

[3.1 Исходные данные 3](#_Toc64299184)

[3.2 Результирующие (выходные) данные 3](#_Toc64299185)

[4 Используемые классы и методы 3](#_Toc64299186)

[5 Текст программы 4](#_Toc64299187)

[5.1 Текст программы 1 4](#_Toc64299188)

[5.2 Текст программы 2 4](#_Toc64299189)

[6 Контрольный пример 5](#_Toc64299190)

[6.1 Контрольный пример для 1 программы 5](#_Toc64299191)

[6.2 Контрольный пример для 2 программы 5](#_Toc64299192)

[7 Полученные результаты 7](#_Toc64299193)

[8 Выводы 8](#_Toc64299194)

[9 Список использованных источников 8](#_Toc64299195)

# Цель работы

Получение практических навыков в создании программы на С#, навыков работы со средствами обработки исключений, навыков работы со средствами конвертации типов и запуска программы из командной строки, а так же умения отладки программы.

# Анализ задачи

Требуется:

1. Написать программу, выводящую на экран приветствие. Вызвать программу-приветствие из командной строки. Отладить программу.
2. Написать программу для деления двух чисел, преобразовать тип переменной, обработать исключения.

# Формальная постановка задачи

## Исходные данные

1. программа:

В качестве исходных данных выступает строка, введенная пользователем.

1. программа:

В качестве исходных данных выступают два целых числа, введенных пользователем.

## Результирующие (выходные) данные

1. программа:

Пользователю выводится приветственное сообщение: «Hello, <строка, введенная пользователем>».

1. программа:

Пользователю в консоль выводится результат деления первого числа на второе или сообщение об ошибке при некорректном вводе или при попытке деления на ноль.

# 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.

1. программа:

В программе определен единственный класс Greeter, в котором объявлен метод Main(), служащий для запуска программы и использующий вышеописанные методы Console.WriteLine() и Console.ReadLine().

1. программа:

В программе определен единственный класс DivideIt, в котором объявлен метод Main(), служащий для запуска программы и использующий методы Console.WriteLine() и Console.ReadLine(). В этом же методе созданы две целочисленные переменные, им присвоены значения, введенные пользователем, и рассмотрены и обработаны исключения, такие как неверный формат введенного числа или деление на ноль.

# Текст программы

## Текст программы 1

1. using System;
2. namespace OOP1Laba\_greeting
3. {
4. class Greeter
5. {
6. static void Main(string[] args)
7. {
8. string my\_name;
9. Console.WriteLine("Please, write your name:");
10. my\_name = Console.ReadLine();
11. Console.WriteLine("Hello, {0}", my\_name);
12. }
13. }
14. }

## Текст программы 2

1. using System;
2. namespace OOP1Laba\_divider
3. {
4. class DivideIt
5. {
6. static void Main(string[] args)
7. {
8. int i, j;
9. string temp;
10. Console.WriteLine("Please, enter first integer number");
11. try
12. {
13. i = Convert.ToInt32(Console.ReadLine());
14. Console.WriteLine("Please, enter second integer number");
15. temp = Console.ReadLine();
16. if (Int32.TryParse(temp, out j))
17. {
18. try
19. {
20. int k = i / j;
21. Console.WriteLine("The result of dividing first on second = {0}", k);
22. }
23. catch (Exception divide\_on\_zero)
24. {
25. if (j == 0)
26. Console.WriteLine("Exception of dividing on zero: {0}", divide\_on\_zero.StackTrace);
27. }
28. }
29. else
30. {
31. Console.WriteLine("Entered string (second integer number) could not be converted to integer");
32. Console.WriteLine("Сannot calculate the result of division");
33. }
34. }
35. catch (Exception wrong\_format)
36. {
37. Console.WriteLine("Exception of wrong format of the number: {0}", wrong\_format.StackTrace);
38. }
39. }
40. }
41. }

# Контрольный пример

## Контрольный пример для 1 программы

На рис. 6.1.1 представлен результат работы программы 1.

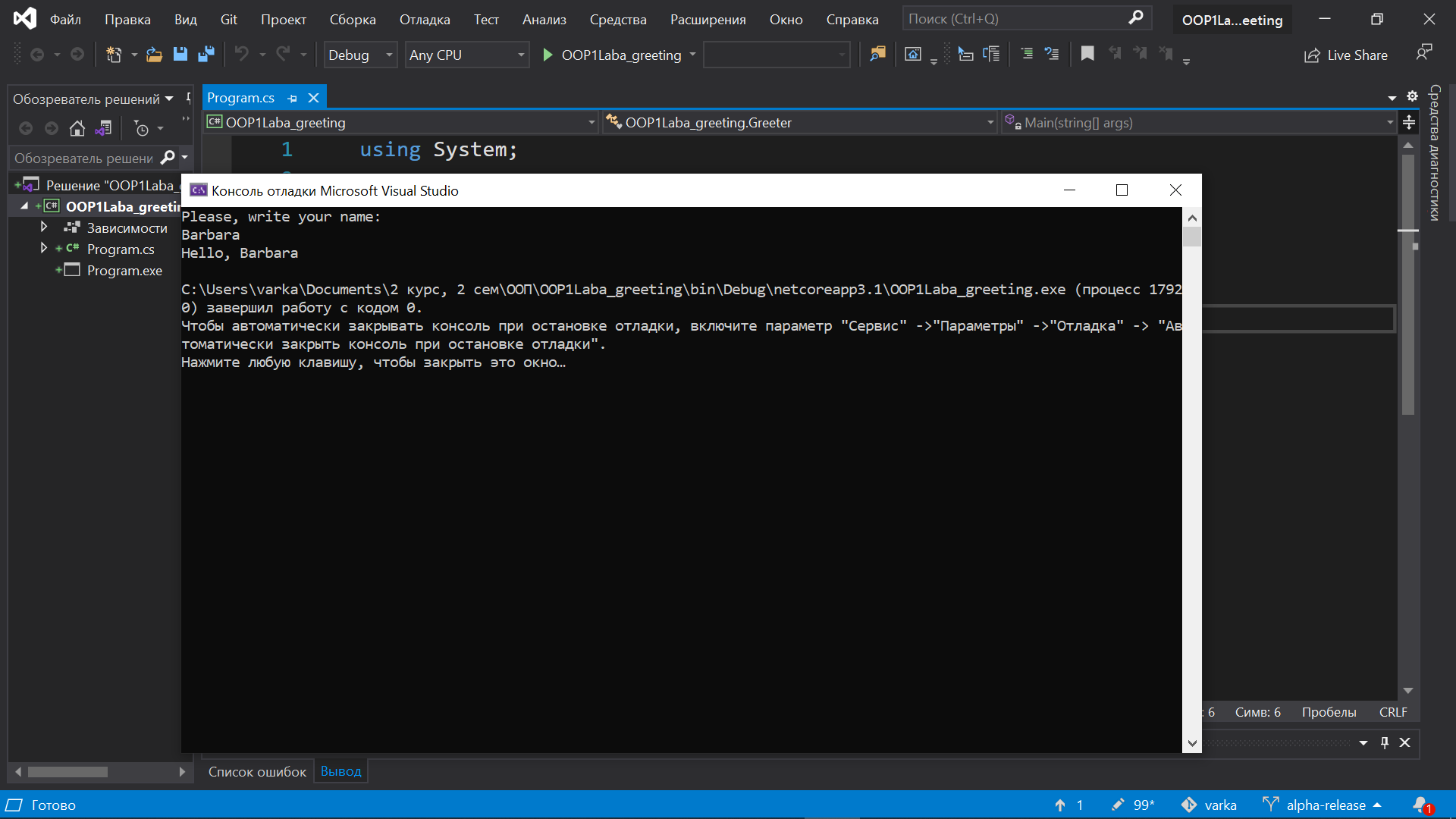


Рис. . — Вывод приветствия пользователю Barbara

## Контрольный пример для 2 программы

На рис. 6.2.1 представлена работа программы с корректными входными данными:

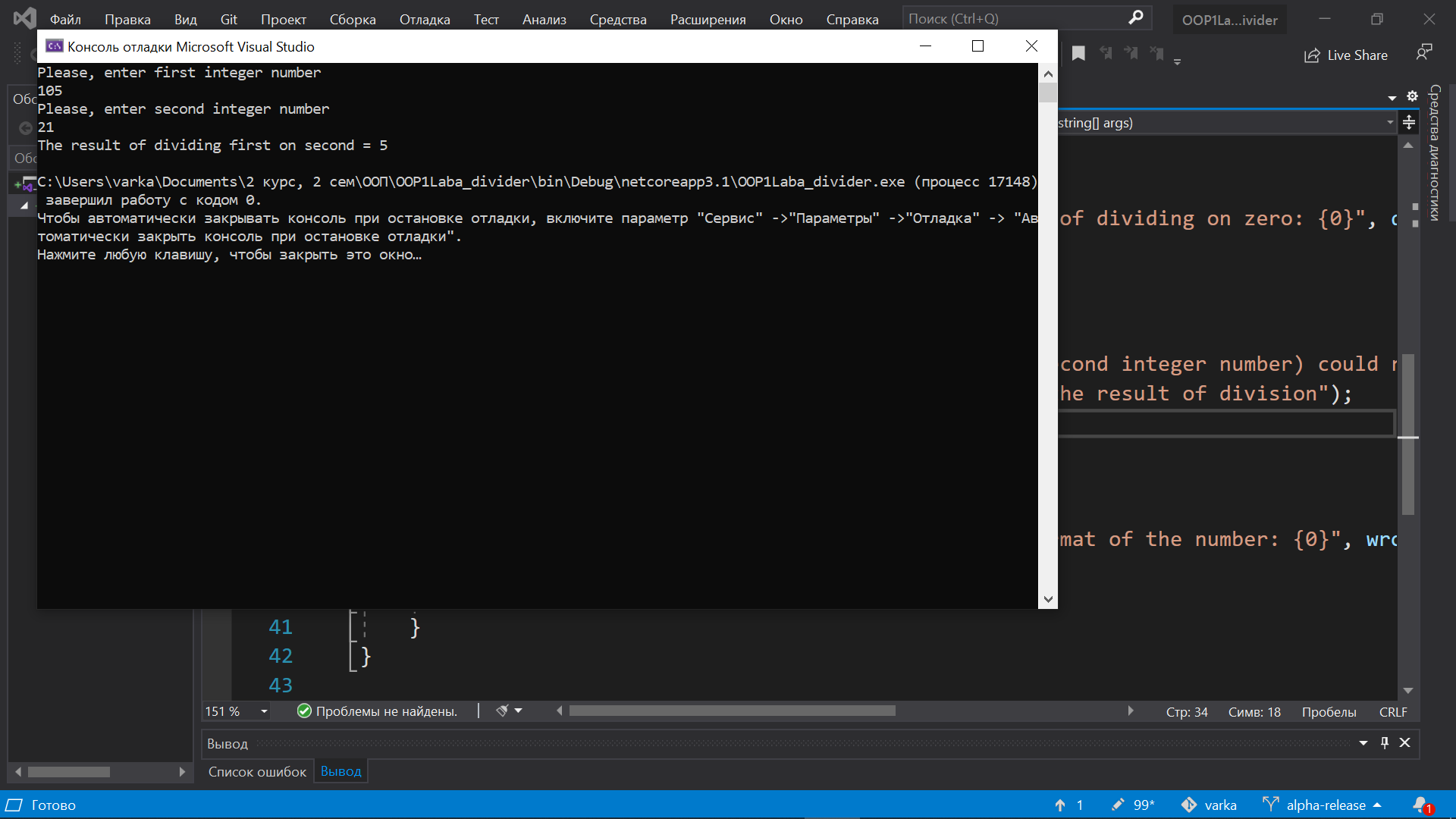


Рис. . — результат выполнения программы 2 с корректными данными

На рис. 6.2.2 представлен результат программы, когда первым ввели число, а вторым аргументом оказалась строка. На рис. 6.2.3 показана аналогичная ситуация, однако неверным является первый аргумент.

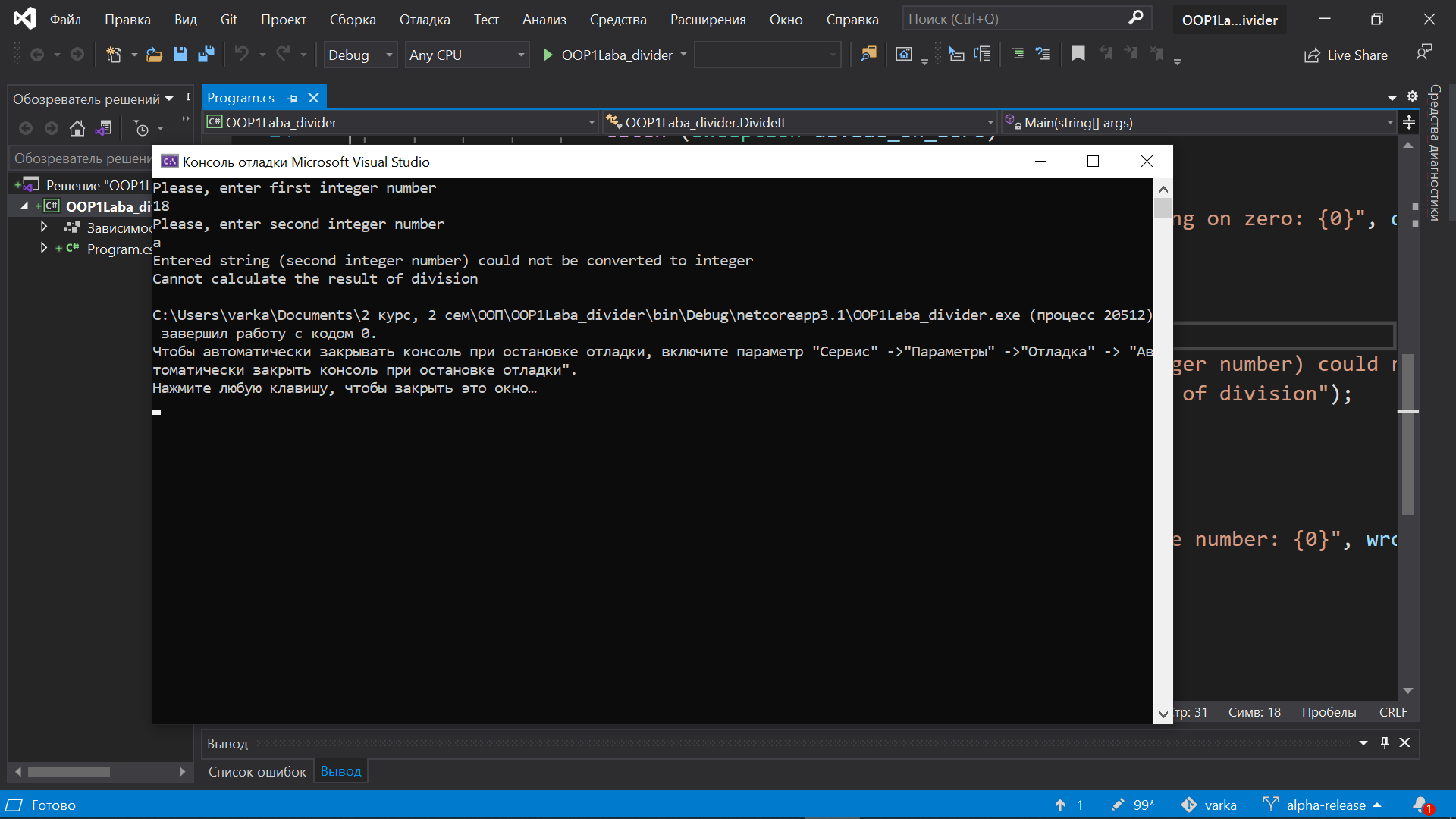


Рис. . — результат работы программы при некорректном втором аргументе

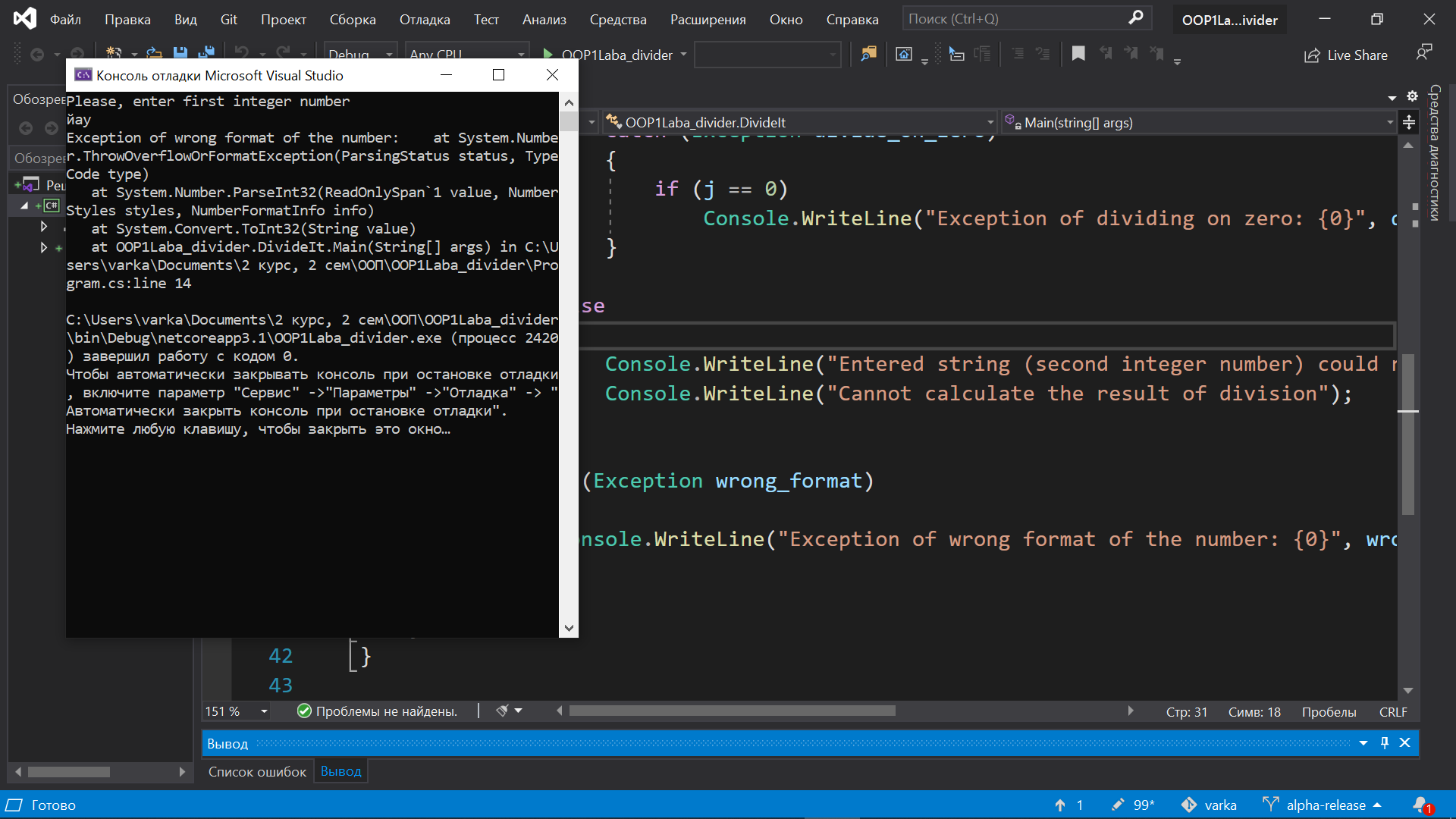


Рис. . — результат работы программы при некорректном первом аргументе

На рис. 6.2.4 представлен результат работы программы 2 при нулевом втором аргументе (деление на ноль):

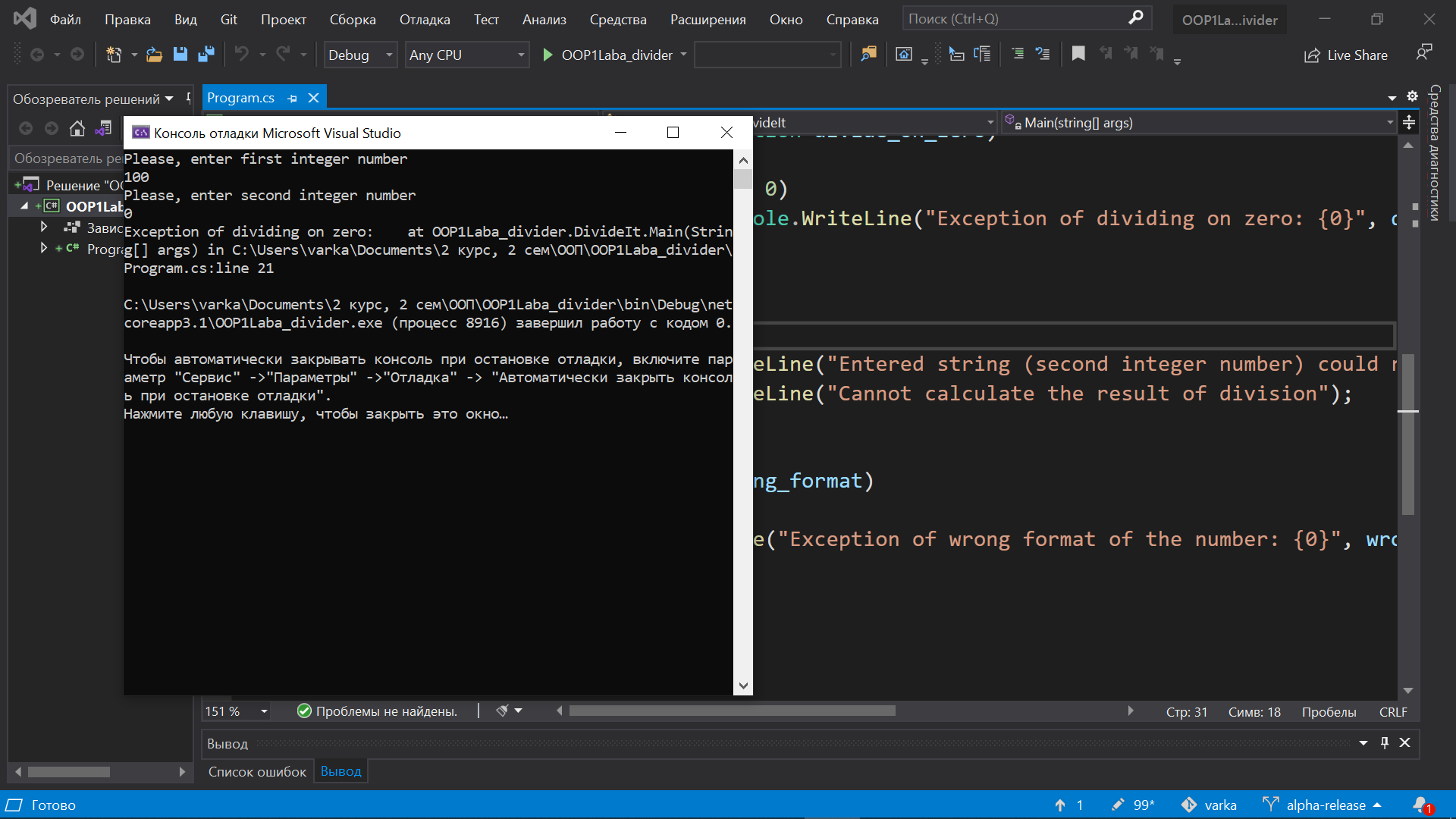


Рис. . — результат работы программы с обработкой исключения при делении на ноль

# Полученные результаты

В результате работы программы 1 на экран было выведено приветственное сообщение: «Hello, Barbara».

Вызов программы 1 из командной строки осуществляется так: сначала в командной строке посредством команды cd нужно перейти в папку, где лежит файл программы Program.cs, затем написать в двойных кавычках полный путь до компилятора csc.exe, не забыв в конце дописать его имя, после выражения в кавычках нужно вызвать Program.cs. Далее нужно вызвать Program.exe, и можно наблюдать выполнение программы прямо в окне командной строки (run terminal), как показано на рис. 7.1.

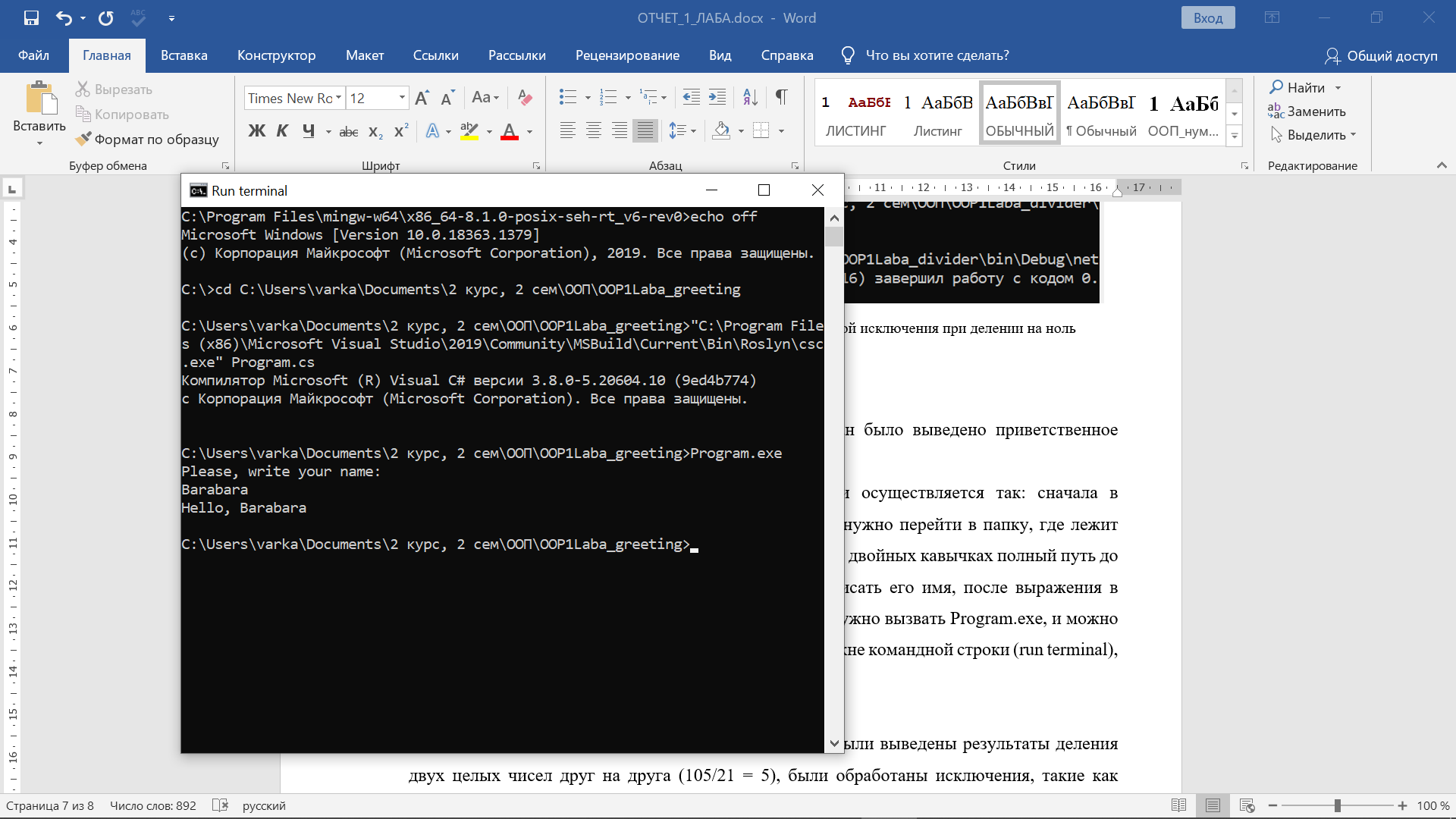


Рис. . — вызов программы 1 из командной строки

Процесс отладки программы показан на рис. 7.2.

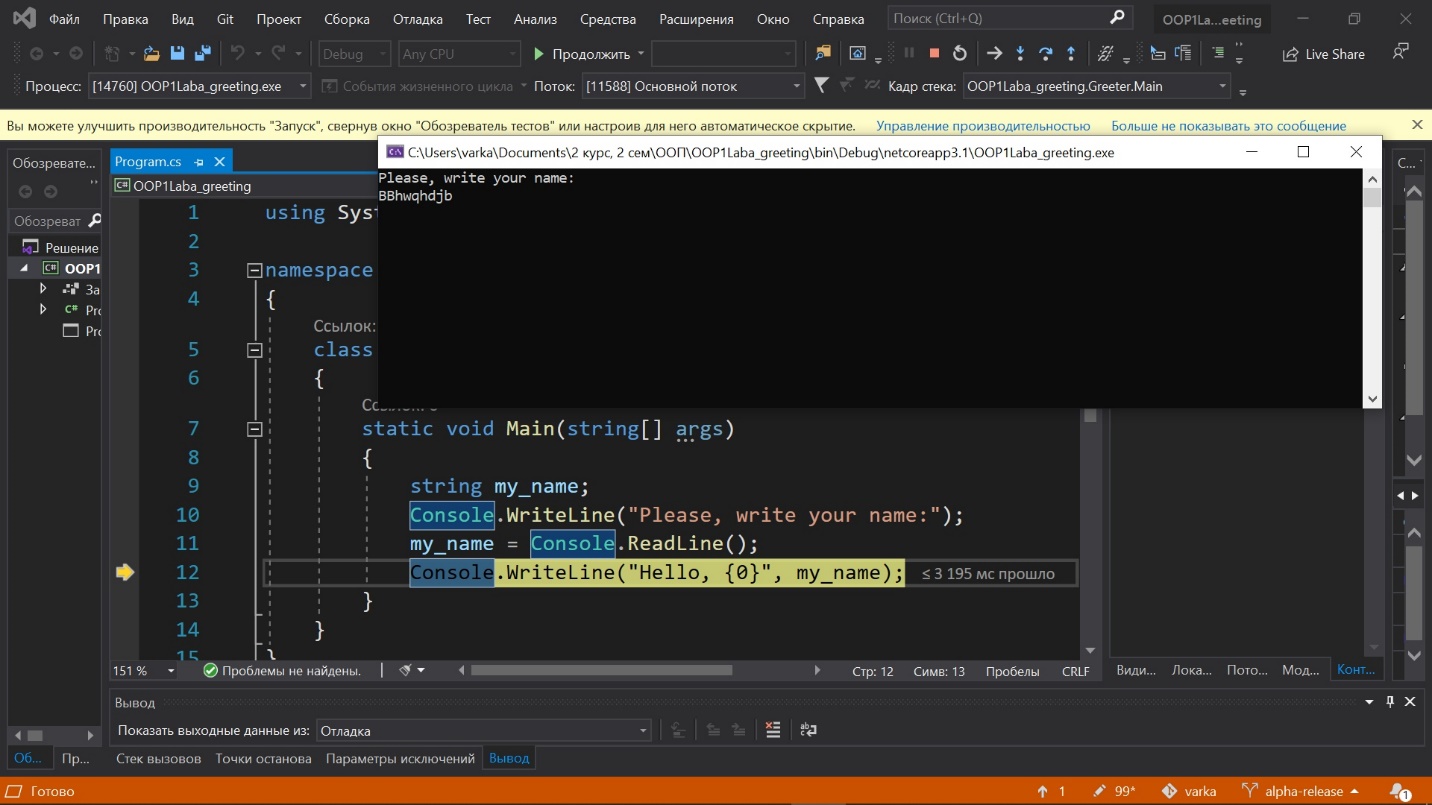


Рис. . — отладка программы 1

В результате работы программы 2 на экран были выведены результаты деления двух целых чисел друг на друга (105/21 = 5), были обработаны исключения, такие как несоответствие аргумента требуемому типу и нулевое значение делителя.

# Выводы

В ходе выполнения данной лабораторной работы:

Были получены знания о написании программ на C#, были изучены

синтаксические правила работы с методами ввода-вывода.

Была усвоена логика построения и синтаксис конструкции для обработки

исключений. Была отмечена важность типа данных в работе с языком C#.

Были получены навыки отладки и запуска программы из командной строки.

# Список использованных источников

Руководство по языку C#:

URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения 14.02.2021)