

Санкт-Петербургский Национальный Исследовательский
Университет Информационных Технологий, Механики и Оптики

Факультет инфокоммуникационных технологий

Лабораторная работа №1
Создание программы с помощью среды разработки Visual
Studio .NET

Выполнил
Стафеев И.А.

Группа
К3221

Проверил
Иванов С.Е.

Санкт-Петербург,
2024

СОДЕРЖАНИЕ

	Стр.
ВВЕДЕНИЕ	3
1 Упражнение 1.....	4
2 Упражнение 2.....	5
3 Упражнение 3.....	7
4 Упражнение 4.....	10
5 Упражнение 5.....	14
ЗАКЛЮЧЕНИЕ	17

ВВЕДЕНИЕ

Цель работы: изучение структуры программы на языке C# и приобретение навыков ее компиляции и отладки.

Для достижения цели необходимо выполнить следующие упражнения:

1. Создание простой программы в текстовом редакторе
2. Создание программы с помощью среды разработки Visual Studio .NET
3. Использование отладчика Visual Studio .NET
4. Добавление в C#-программу обработчика исключительных ситуаций
5. Расчет площади треугольника

1 Упражнение 1

Задача: В этом упражнении вы напишете программу на языке C#, используя обычный текстовый редактор, например, Блокнот. В окне командной строки программа будет спрашивать, как вас зовут, и затем будет приветствовать вас по имени.

Код программы приведен на рисунке 1.

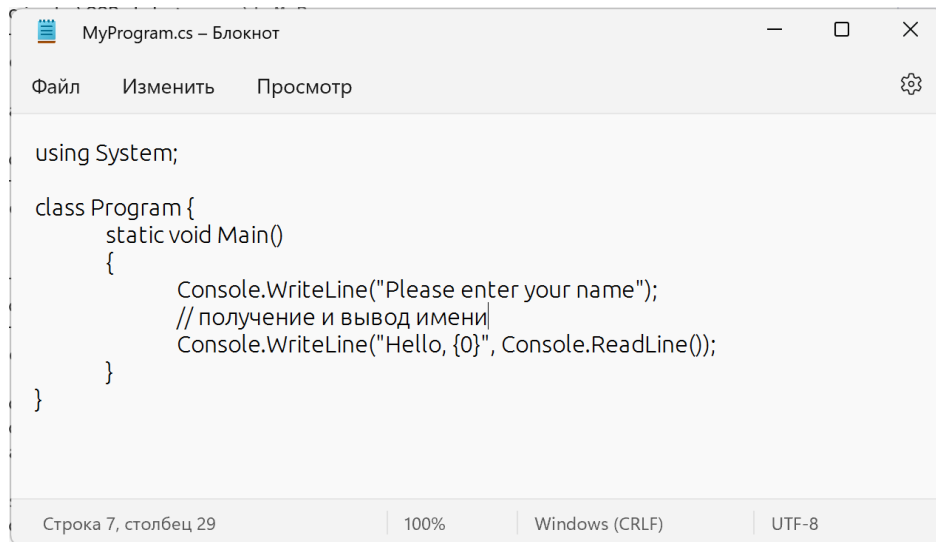


Рисунок 1 — Код программы, приветствующей пользователя

Процесс компиляции программы и ее запуска через командную строку приведен на рисунке 2.

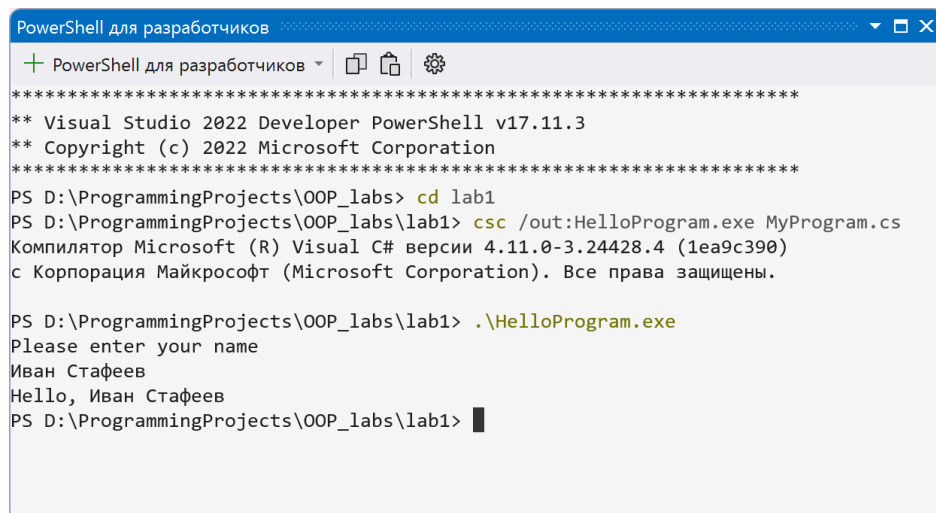


Рисунок 2 — Компиляция и запуск программы

2 Упражнение 2

Задача: В этом упражнении вы напишете программу первого упражнения, используя среду разработки Visual Studio.

Шаги создания нового проекта приведены на рисунке 3.

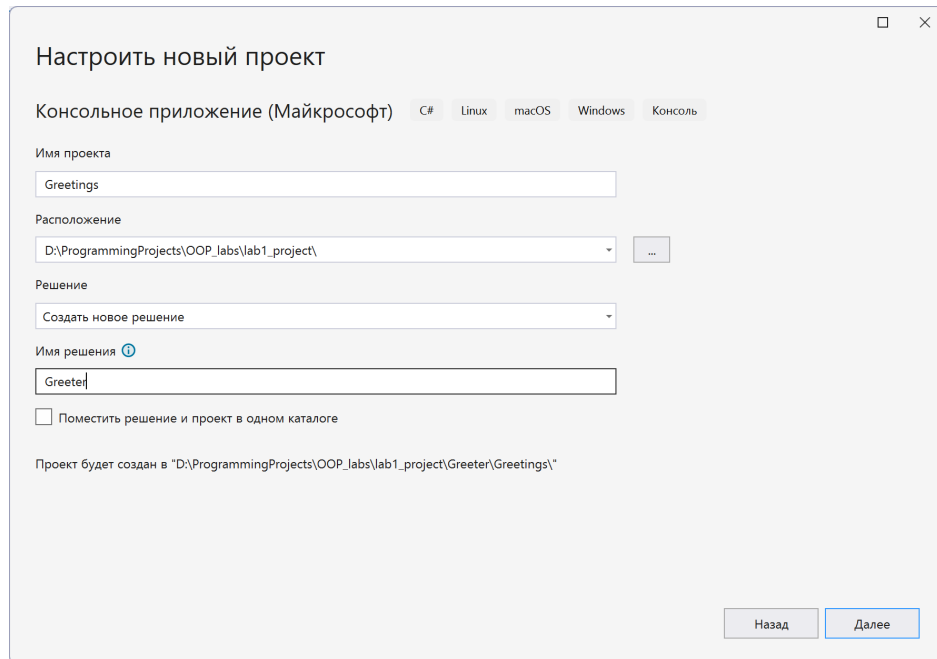


Рисунок 3 — Создание проекта в Visual Studio

Код программы и результат компиляции приведен на рисунке 4.

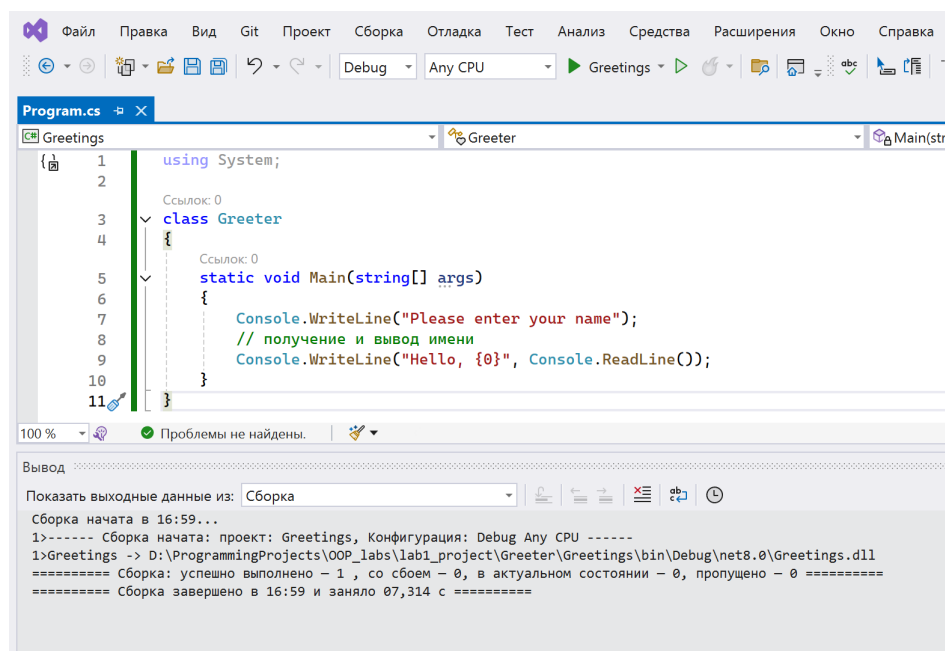
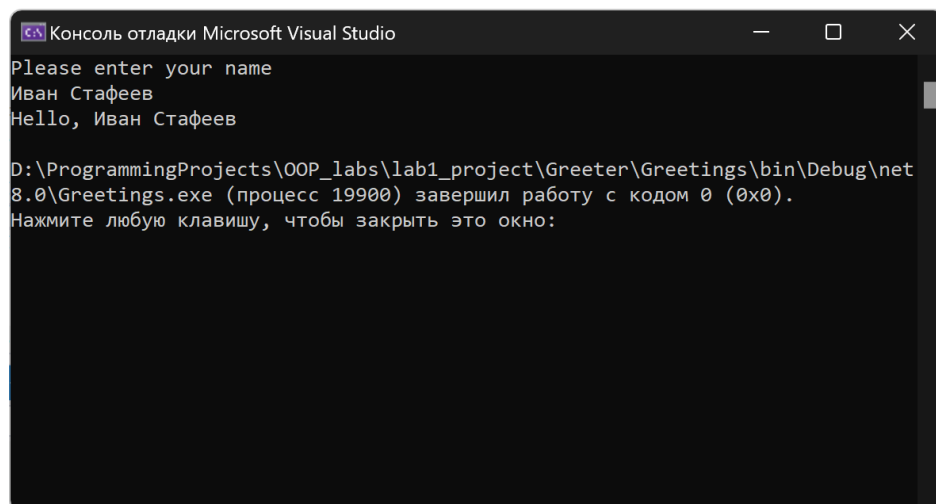


Рисунок 4 — Написание кода программы и ее компиляция

Процесс выполнения написанной программы показан на рисунке 5.



```
Консоль отладки Microsoft Visual Studio
Please enter your name
Иван Стафеев
Hello, Иван Стафеев

D:\ProgrammingProjects\OOP_labs\lab1_project\Greeter\Greetings\bin\Debug\net
8.0\Greetings.exe (процесс 19900) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 5 — Запуск написанной программы

3 Упражнение 3

Задача: В этом задании вы научитесь работать с интегрированным отладчиком Visual Studio .NET: проходить программу по шагам и просматривать значения переменных.

Код программы, добавление точки остановки и результат компиляции приведен на рисунке 6.

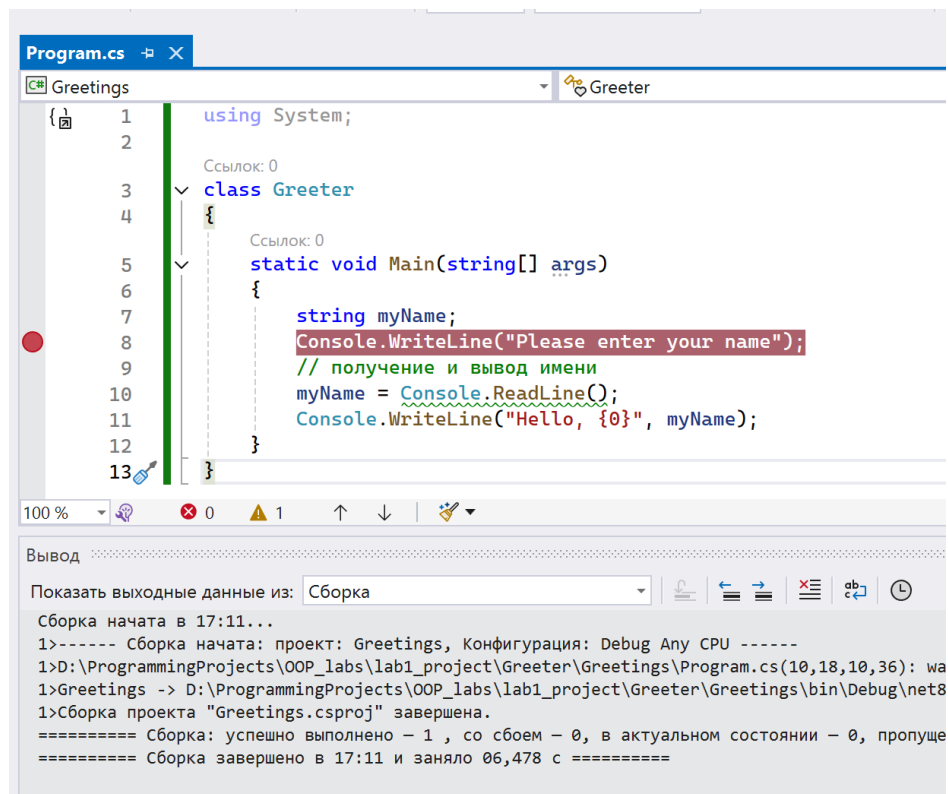


Рисунок 6 — Написания кода программы и создание точки остановки

Процесс отладки программы показан на рисунках 7, 8 и 9.

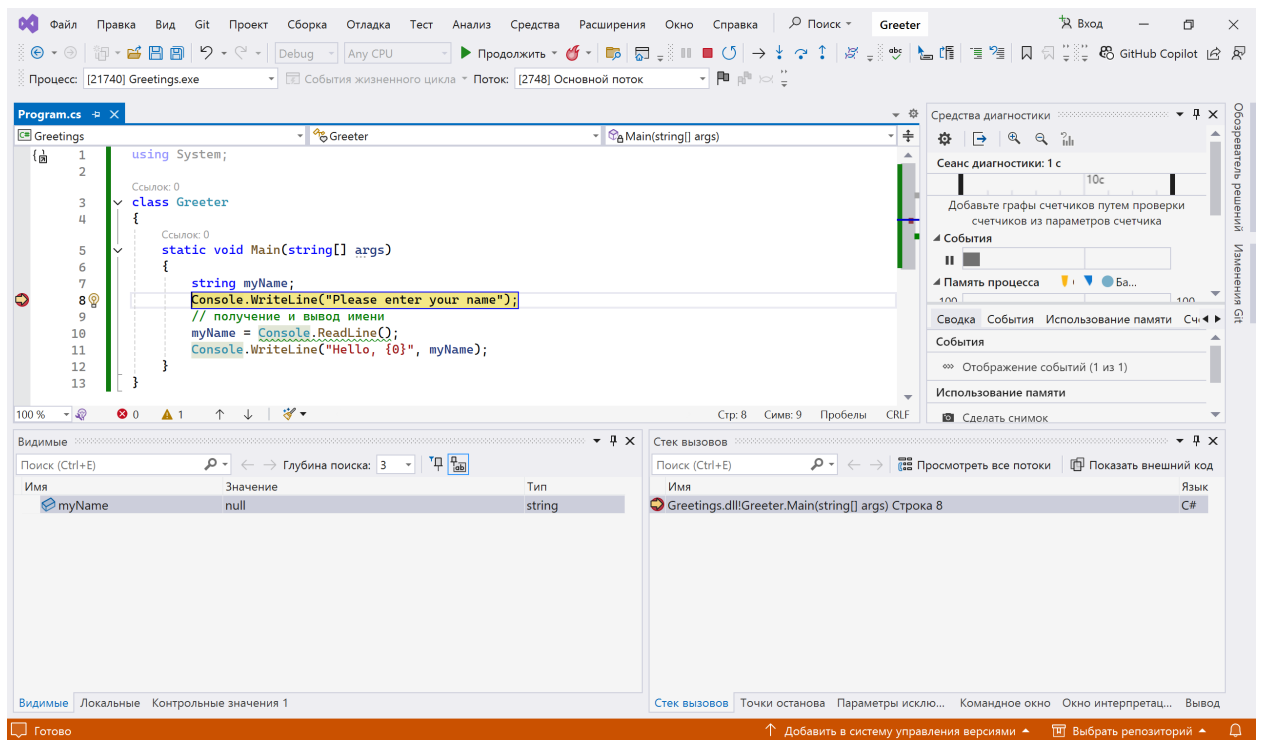


Рисунок 7 — Значение переменной на первом шаге отладки

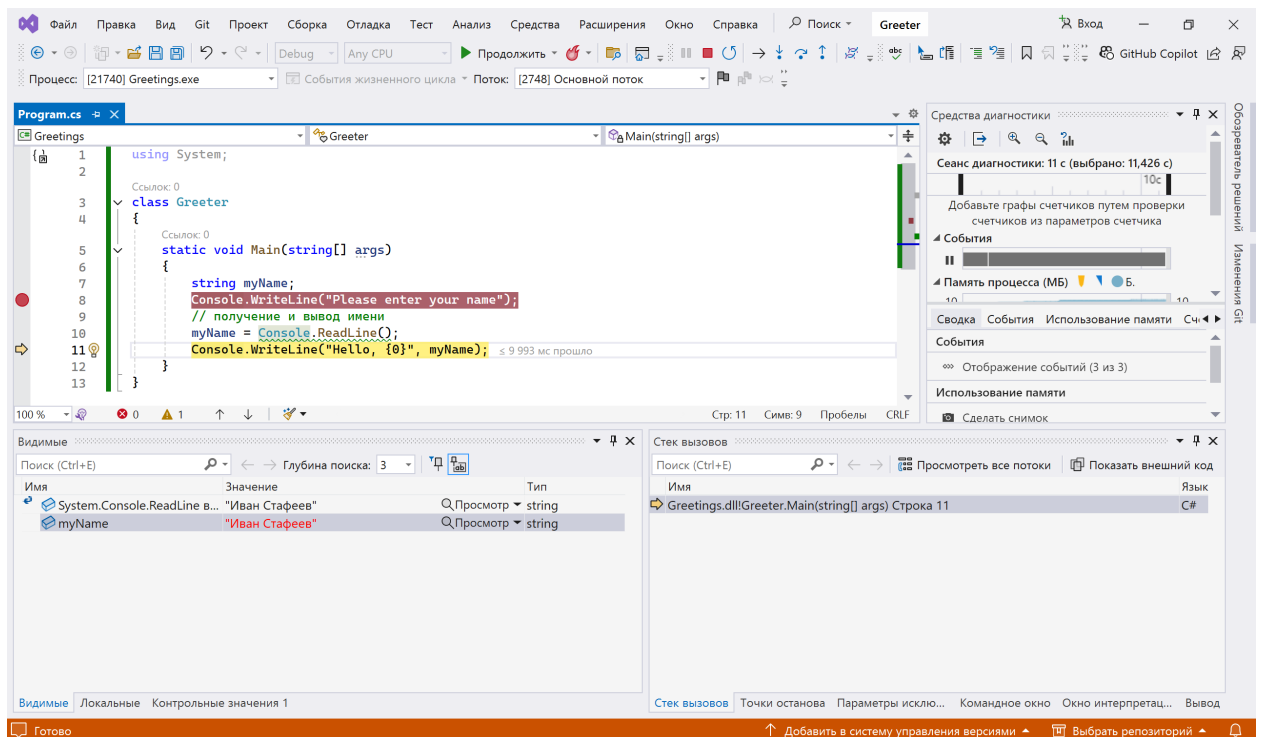


Рисунок 8 — Значение переменной на третьем шаге отладки

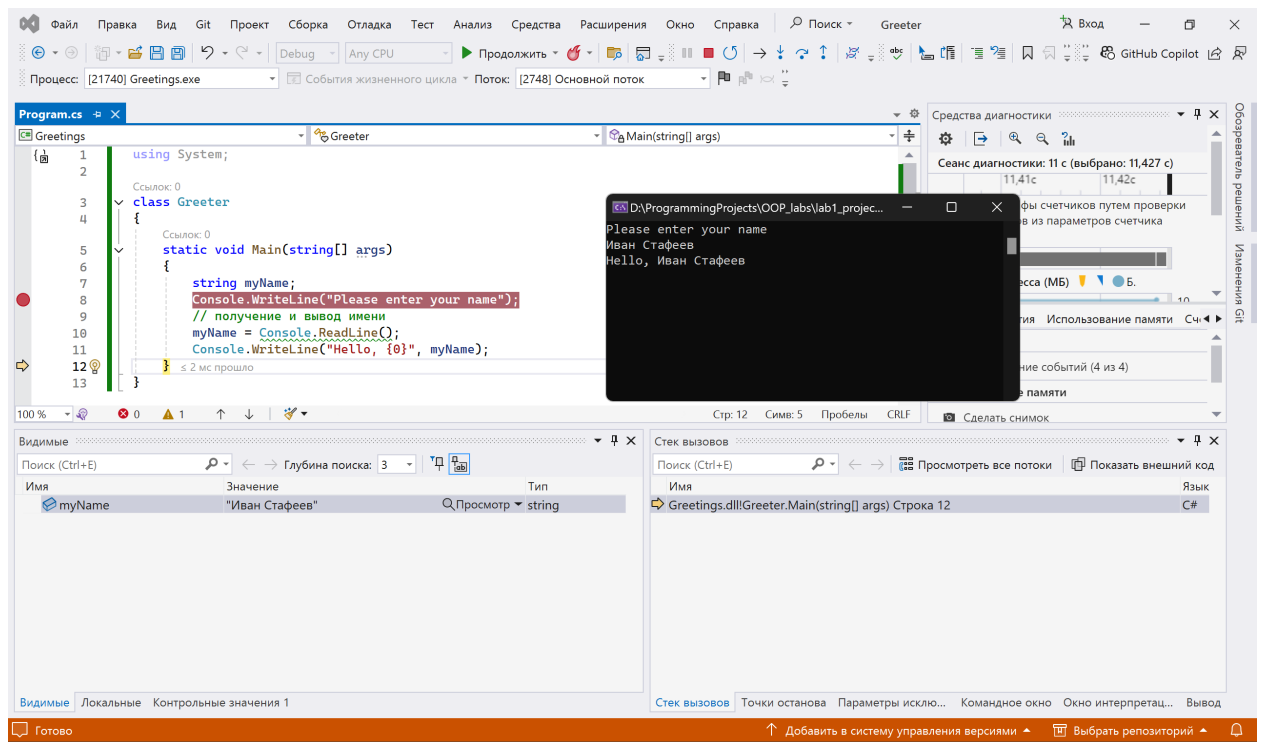


Рисунок 9 — Результат выполнения программы

4 Упражнение 4

Задача: В этом упражнении вы напишете программу, в которой будет использоваться обработчик исключительных ситуаций, который будет отлавливать ошибки времени выполнения. Программа будет запрашивать у пользователя два целых числа, делить первое число на второе и выводить полученный результат.

Код первоначальной программы приведен на рисунке 10.

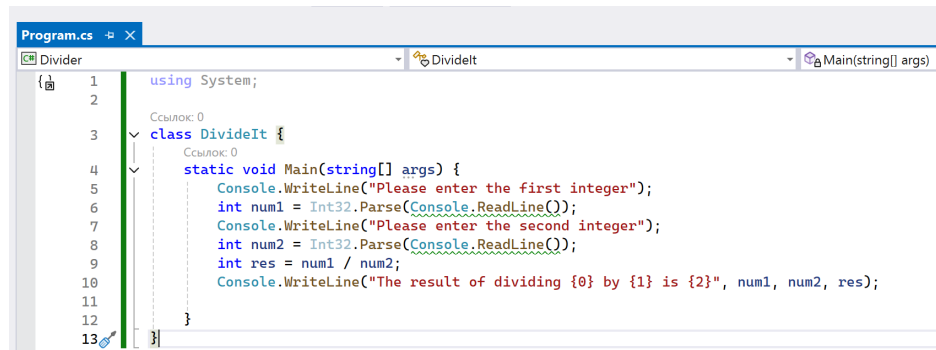


Рисунок 10 — Написания кода программы

При выполнении программы возможны три случая: успешное выполнение (11), выполнение с ошибкой о неверном формате входных данных (12) и выполнение с ошибкой при делении на ноль (13).

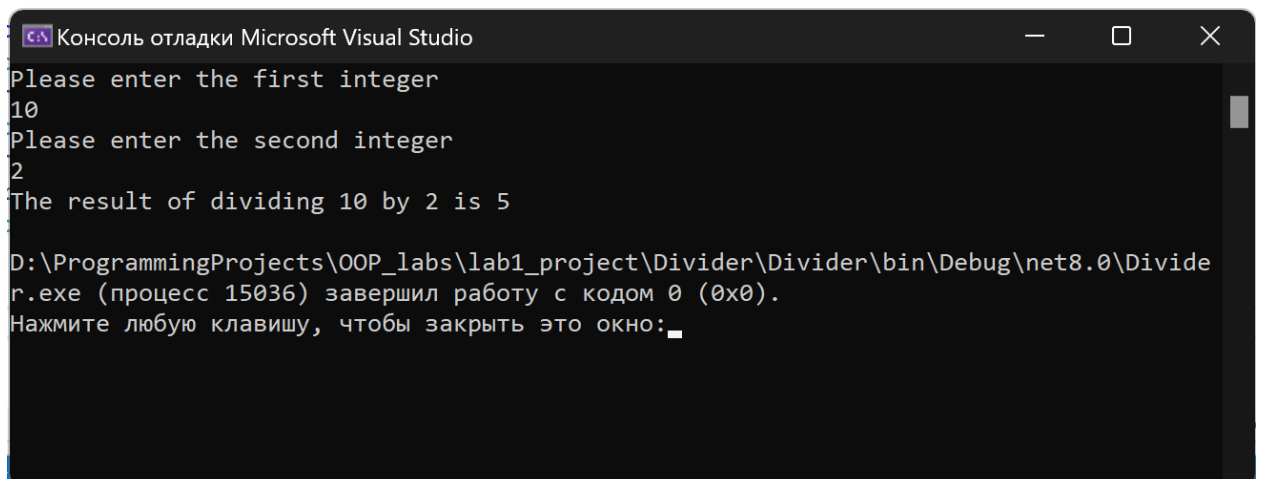
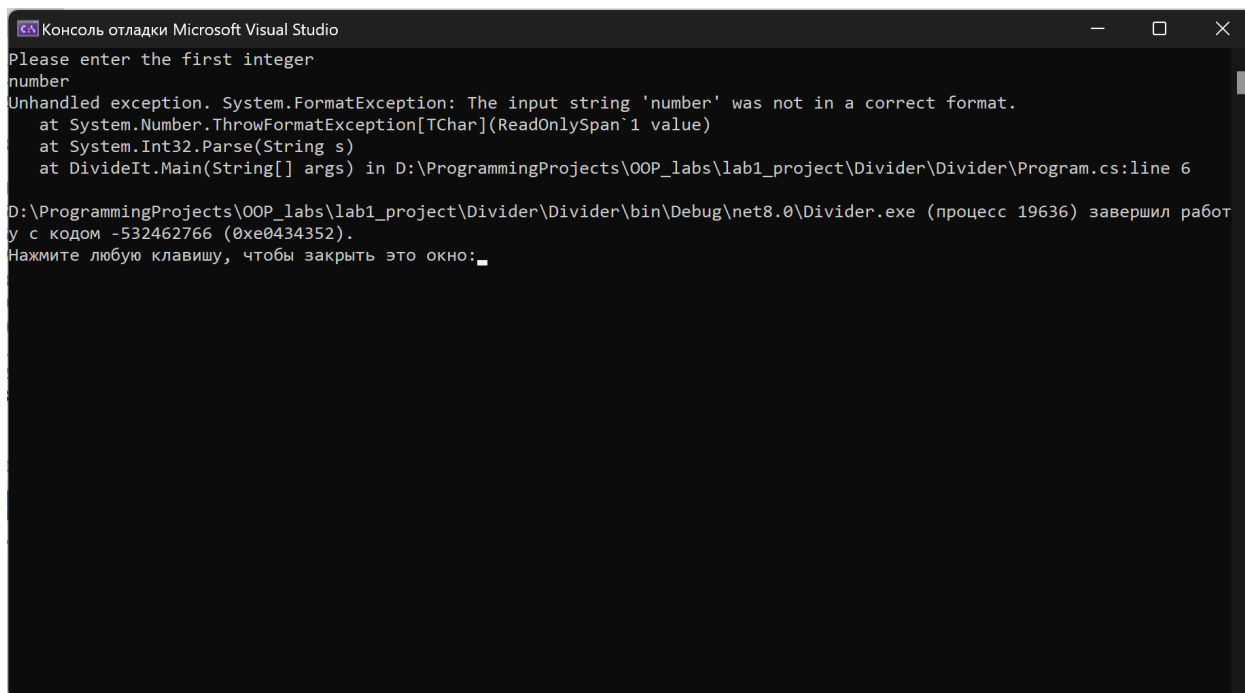


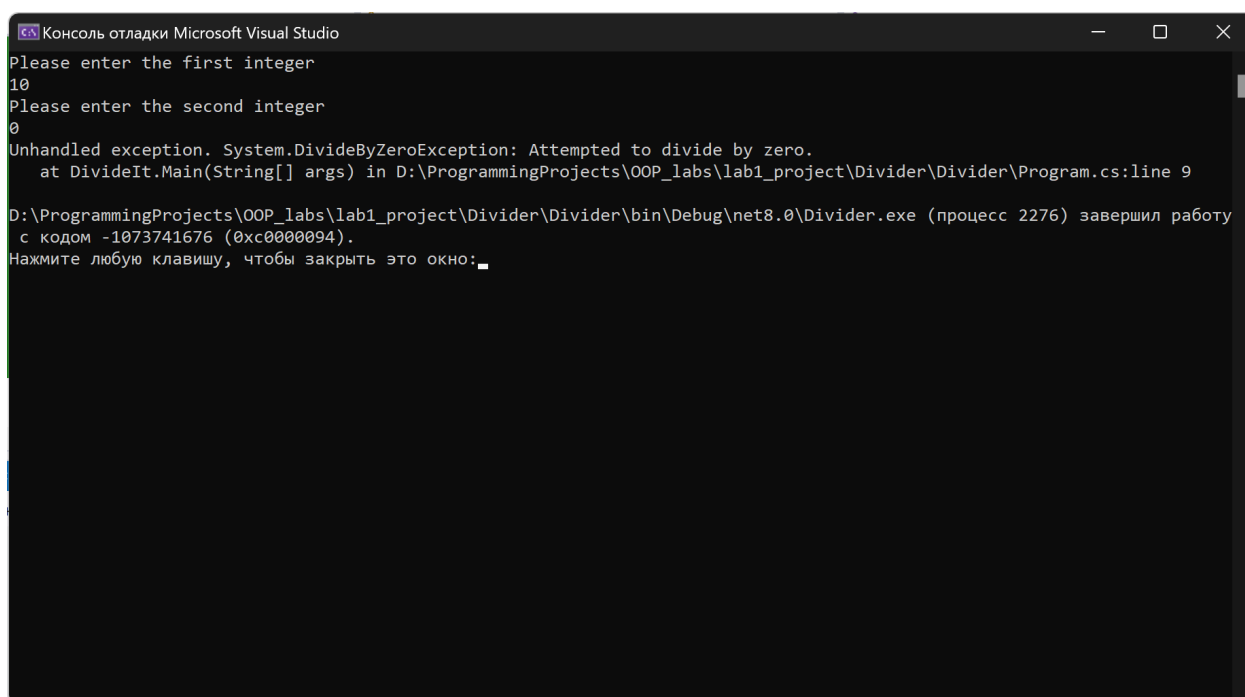
Рисунок 11 — Успешное выполнение программы



```
Консоль отладки Microsoft Visual Studio
Please enter the first integer
number
Unhandled exception. System.FormatException: The input string 'number' was not in a correct format.
    at System.Number.ThrowFormatException[TChar](ReadOnlySpan`1 value)
    at System.Int32.Parse(String s)
    at DivideIt.Main(String[] args) in D:\ProgrammingProjects\OOP_labs\lab1_project\Divider\Divider\Program.cs:line 6

D:\ProgrammingProjects\OOP_labs\lab1_project\Divider\Divider\bin\Debug\net8.0\Divider.exe (процесс 19636) завершил работу
с кодом -532462766 (0xe0434352).
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок 12 — Ошибка при неверном формате входных данных



```
Консоль отладки Microsoft Visual Studio
Please enter the first integer
10
Please enter the second integer
0
Unhandled exception. System.DivideByZeroException: Attempted to divide by zero.
    at DivideIt.Main(String[] args) in D:\ProgrammingProjects\OOP_labs\lab1_project\Divider\Divider\Program.cs:line 9

D:\ProgrammingProjects\OOP_labs\lab1_project\Divider\Divider\bin\Debug\net8.0\Divider.exe (процесс 2276) завершил работу
с кодом -1073741676 (0xc0000094).
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок 13 — Ошибка при делении на ноль

Для успешного выполнения программы во всех случаях были написаны исключения. Измененный код программы приведен на рисунке 14.

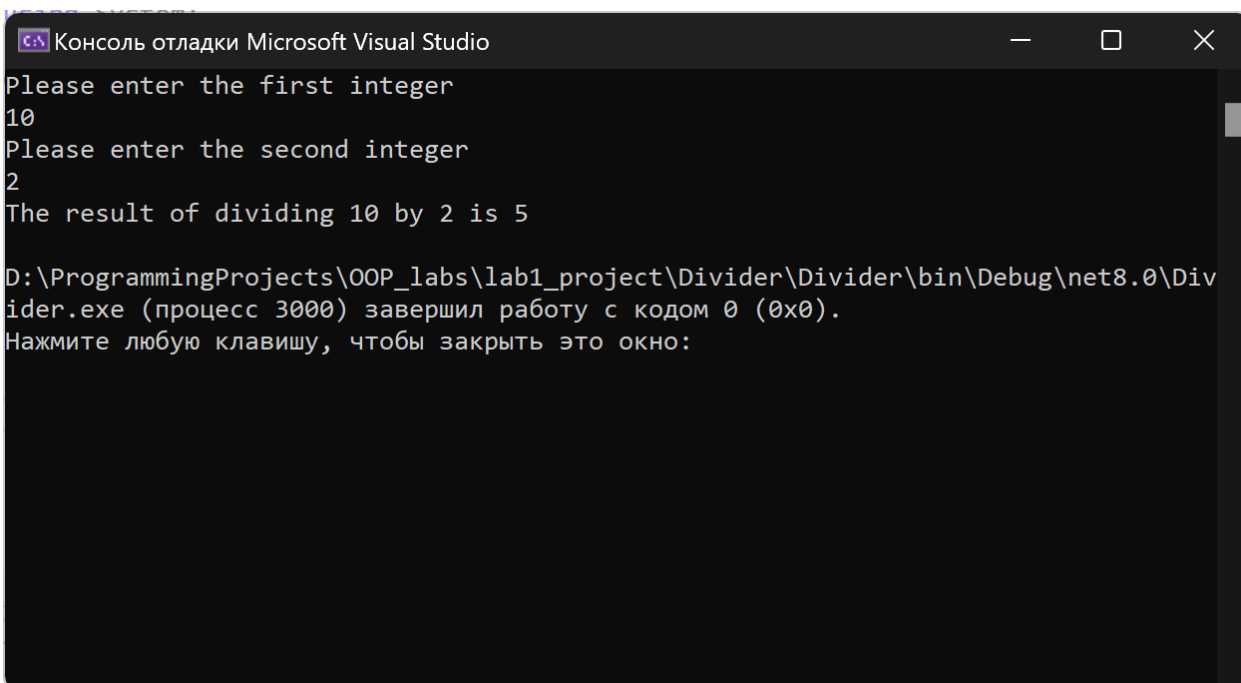
```

1  using System;
2
3  class DivideIt {
4      static void Main(string[] args) {
5          try
6          {
7              Console.WriteLine("Please enter the first integer");
8              int num1 = Int32.Parse(Console.ReadLine());
9              Console.WriteLine("Please enter the second integer");
10             int num2 = Int32.Parse(Console.ReadLine());
11             int res = num1 / num2;
12             Console.WriteLine("The result of dividing {0} by {1} is {2}", num1, num2, res);
13         }
14         catch (FormatException e)
15         {
16             Console.WriteLine("A format exception was thrown: {0}", e.Message);
17         }
18         catch (DivideByZeroException e) {
19             Console.WriteLine("A value exception was thrown: {0}", e.Message);
20         }
21         catch (Exception e) {
22             Console.WriteLine("An exception was thrown: {0}", e.Message);
23         }
24     }
25 }
26

```

Рисунок 14 — Код программы с обработкой ошибок

Результаты выполнения программы при тех же входных данных, что и в первых трех тестах, показаны на рисунках 15, 16 и 17.



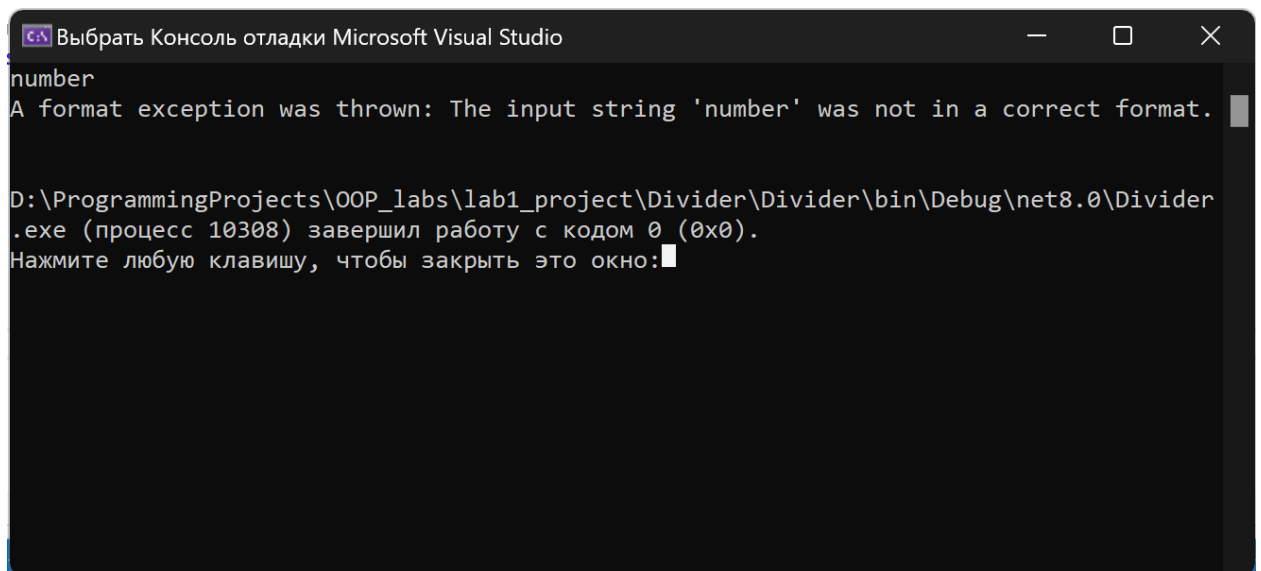
```

Консоль отладки Microsoft Visual Studio
Please enter the first integer
10
Please enter the second integer
2
The result of dividing 10 by 2 is 5

D:\ProgrammingProjects\OOP_labs\lab1_project\Divider\Divider\bin\Debug\net8.0\Div
ider.exe (процесс 3000) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:

```

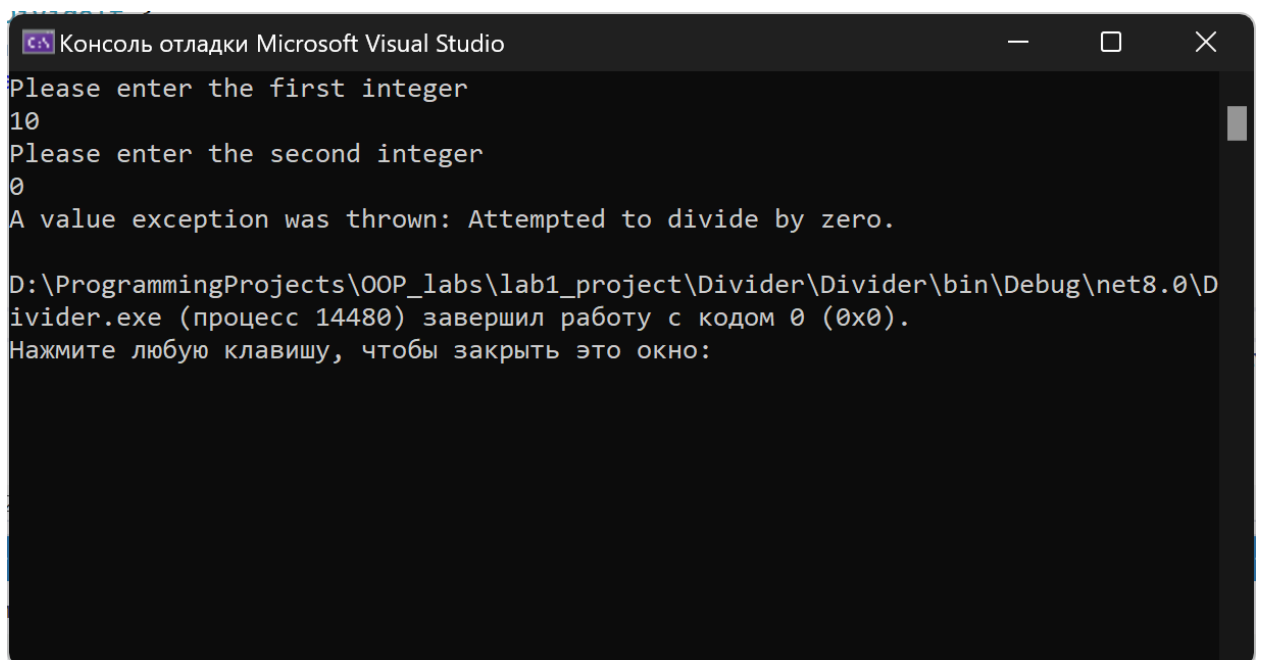
Рисунок 15 — Успешное выполнение программы



```
Выбрать Консоль отладки Microsoft Visual Studio
number
A format exception was thrown: The input string 'number' was not in a correct format.

D:\ProgrammingProjects\OOP_labs\lab1_project\Divider\Divider\bin\Debug\net8.0\Divider
.exe (процесс 10308) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 16 — Выполнение с обработкой ошибки неверного формата данных



```
Консоль отладки Microsoft Visual Studio
Please enter the first integer
10
Please enter the second integer
0
A value exception was thrown: Attempted to divide by zero.

D:\ProgrammingProjects\OOP_labs\lab1_project\Divider\Divider\bin\Debug\net8.0\Div
ider.exe (процесс 14480) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 17 — Выполнение с обработкой ошибки деления на ноль

5 Упражнение 5

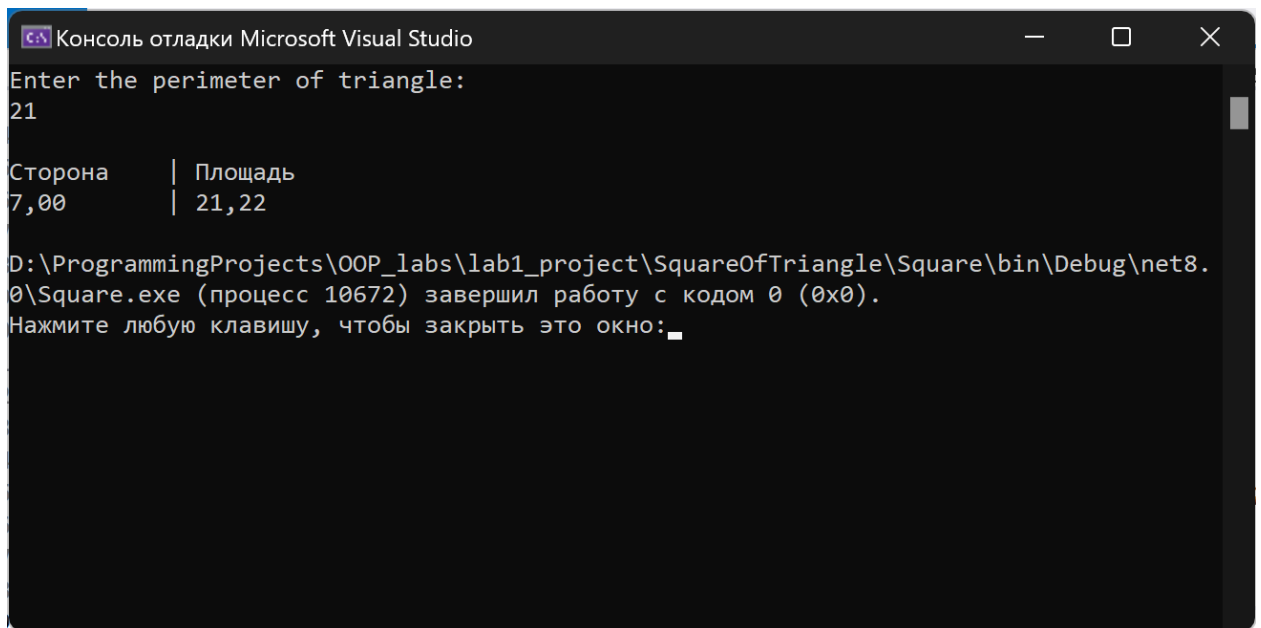
Задача: Требуется создать программу, которая подсчитывает площадь равностороннего треугольника, периметр которого известен.

Программа, которую требуется реализовать, должна учитывать возможные исключения, возникающие при работе, а также корректно обрабатывать входные данные. Поэтому если пользователь вводит как периметр не положительное число или вообще не число, то выводится соответствующее сообщение. Иные ошибки также выводятся. Вывод таблицы производится с использованием форматирования строк. Код программы можно увидеть на рисунке 18.

```
1  using System;
2
3  class SquareOfTriange {
4      static void Main(string[] args) {
5          try {
6              Console.WriteLine("Enter the perimeter of triangle:");
7              double perimeter = double.Parse(Console.ReadLine());
8              if (perimeter <= 0) { // периметр может быть только положительным
9                  throw new FormatException();
10             }
11             // вычисление площади
12             double square = perimeter * perimeter * Math.Sqrt(3) / 36;
13             // вывод таблицы с результатами
14             Console.WriteLine("\n{0,-10} | {1,-10}", "Сторона", "Площадь");
15             Console.WriteLine("{0,-10:f2} | {1,-10:f2}", perimeter / 3, square);
16         }
17         catch (FormatException e) // обработка неверных входных данных
18         {
19             Console.WriteLine("The input data must be a positive number");
20         }
21         catch (Exception e)
22         {
23             Console.WriteLine("The unexpected error has occurred: {0}", e.Message);
24         }
25     }
26 }
```

Рисунок 18 — Код программы для вычисления площади равностороннего треугольника

Примеры выполнения программы для корректных данных и для нечислового, неположительного формата входных данных приведены на рисунках 19, 20 и 21.



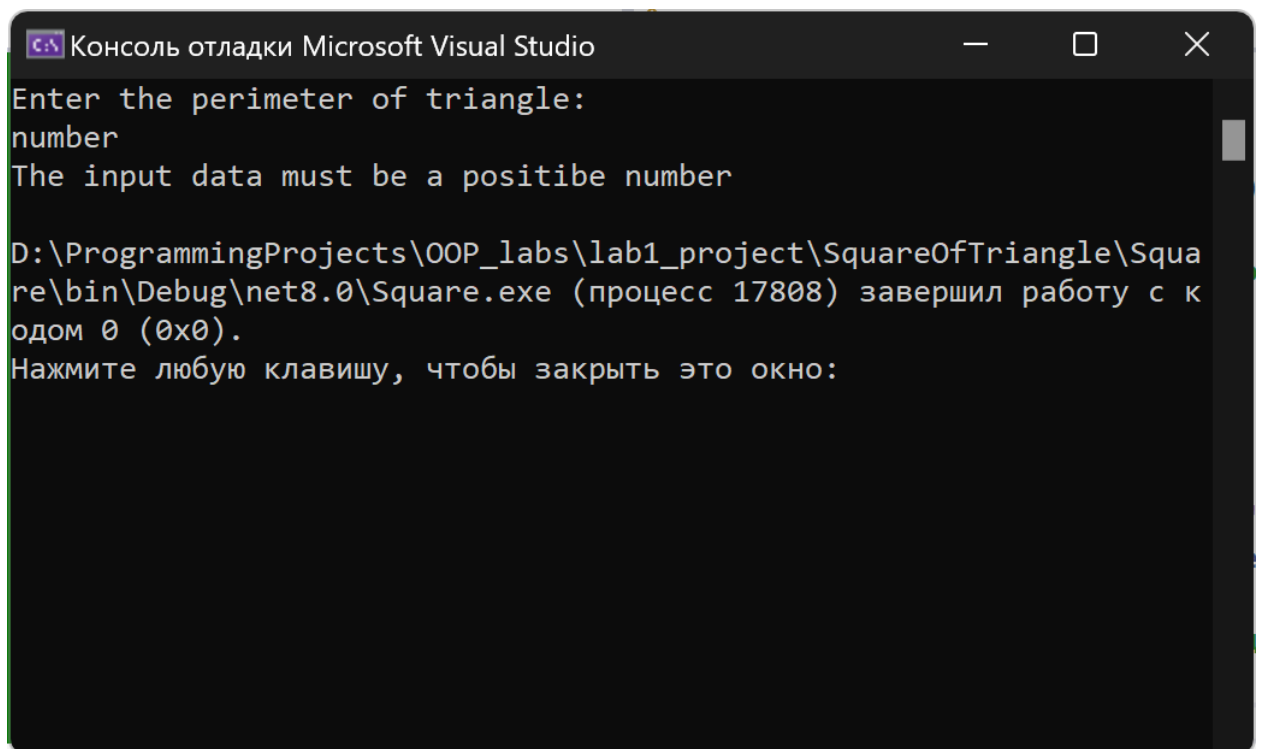
```
Консоль отладки Microsoft Visual Studio
Enter the perimeter of triangle:
21

Сторона    | Площадь
7,00      | 21,22

D:\ProgrammingProjects\OOP_labs\lab1_project\SquareOfTriangle\Square\bin\Debug\net8.0\Square.exe (процесс 10672) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:

```

Рисунок 19 — Результат выполнения программы с корректными данными

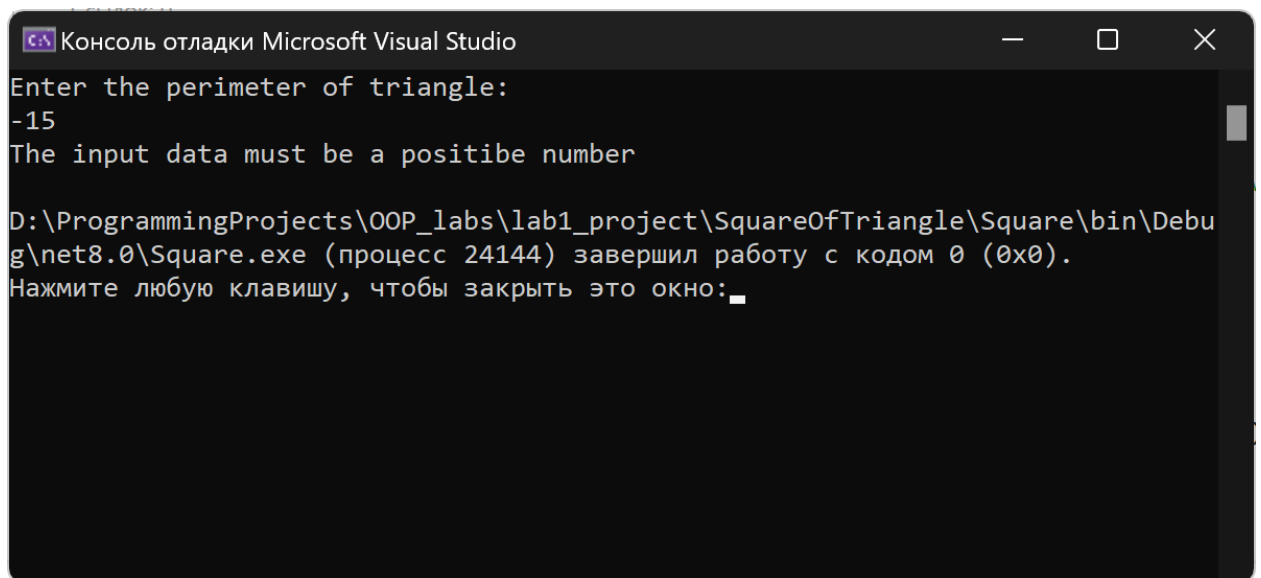


```
Консоль отладки Microsoft Visual Studio
Enter the perimeter of triangle:
number
The input data must be a positibe number

D:\ProgrammingProjects\OOP_labs\lab1_project\SquareOfTriangle\Square\bin\Debug\net8.0\Square.exe (процесс 17808) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:

```

Рисунок 20 — Обработка ошибки при нечисловом формате данных



```
Консоль отладки Microsoft Visual Studio
Enter the perimeter of triangle:
-15
The input data must be a positive number

D:\ProgrammingProjects\OOP_labs\lab1_project\SquareOfTriangle\Square\bin\Debug\net8.0\Square.exe (процесс 24144) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Рисунок 21 — Обработка ошибки, когда введено неположительное число

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы были выполнены все требуемые упражнения. Цель работы достигнута. Получены знания о структуре программы на языке C# и получены навыки ее компиляции и отладки.