

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ITMO University**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**По дисциплине** Объектно-ориентированное программирование

**Тема работы** Создание программы с помощью среды разработки Visual  
Studio.NET

**Обучающийся** Крестьянова Елизавета Федоровна

**Факультет** факультет инфокоммуникационных технологий

**Группа** K3223

**Направление подготовки** 11.03.02 Инфокоммуникационные технологии и  
системы связи

**Образовательная программа** Программирование в  
инфокоммуникационных системах

<b>Обучающийся</b>	_____	_____	<u>Крестьянова Е.Ф.</u>
	(дата)	(подпись)	(Ф.И.О.)

<b>Руководитель</b>	_____	_____	<u>Иванов С.Е.</u>
	(дата)	(подпись)	(Ф.И.О.)

## СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ .....	3
1 Упражнение №1. Создание простой программы в текстовом редакторе .....	4
2 Упражнение №2. Создание программы с помощью среды разработки Visual Studio .NET.....	6
3 Упражнение №3. Использование отладчика Visual Studio .NET.....	8
4 Упражнение №4. Добавление в C#-программу обработчика исключительных ситуаций.....	10
5 Упражнение №5. Расчёт площади треугольника.....	14
ЗАКЛЮЧЕНИЕ .....	16

## **ВВЕДЕНИЕ**

В данном отчёте представлено выполнение лабораторной работы по дисциплине «Объектно-ориентированное программирование».

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

# 1 УПРАЖНЕНИЕ №1. СОЗДАНИЕ ПРОСТОЙ ПРОГРАММЫ В ТЕКСТОВОМ РЕДАКТОРЕ

В этом упражнении стоит задача написания программы на языке C# в обычном текстовом редакторе. Программа должна спрашивать имя пользователя и приветствовать его в окне командной строки.

В блокноте был написан код простой программы, принимающей ввод пользователя и приветствующей его. Код можно увидеть на рисунке 1.1.

Файл был сохранён под названием "MyProgram.cs".

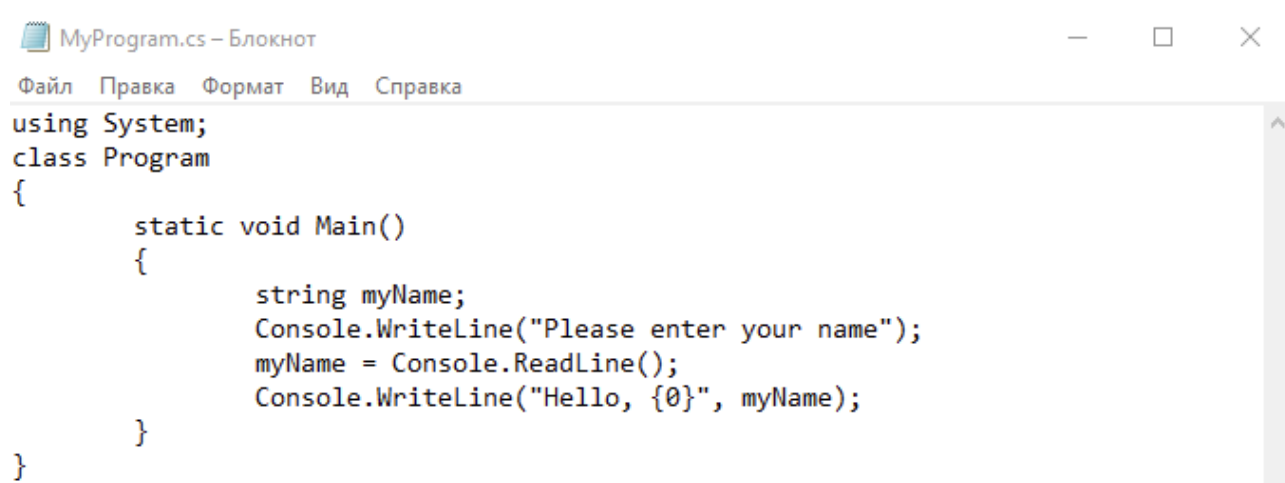


Рисунок 1.1 — Упр №1: Программа в блокноте

Затем было установлено ПО Visual Studio. В его окне открывается консоль cmd, что продемонстрировано на рисунке 1.2. С помощью неё можно компилировать написанные программы.

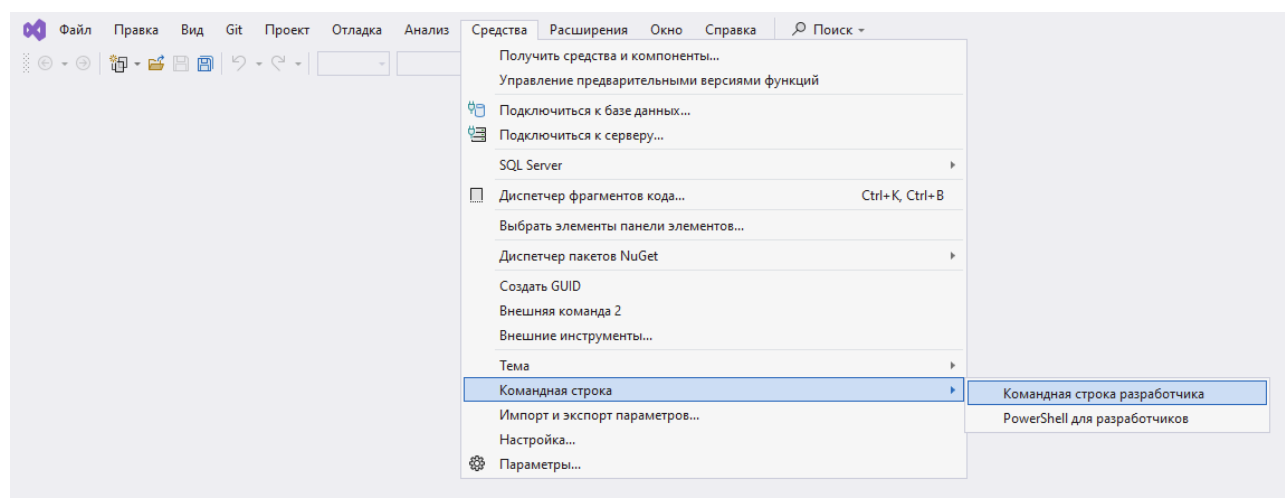
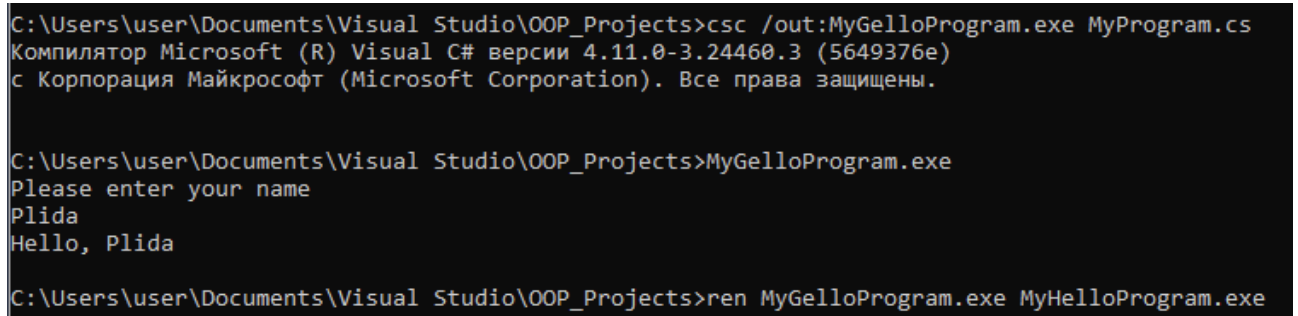


Рисунок 1.2 — Упр №1: Открытие консоли

Перейдя в папку с созданным файлом «MyProgram.cs», была введена команда «csc /out:MyGelloProgram.exe MyProgram.cs», что создало .exe файл на основе написанного кода. Затем программа была запущена через простую команду «MyGelloProgram».

На рисунке 1.3 показаны процесс компиляции, ввод имени и вывод программы; а также команда переименования файлов, полезная для исправления случайных опечаток.



```
C:\Users\user\Documents\Visual Studio\OOP_Projects>csc /out:MyGelloProgram.exe MyProgram.cs
Компилятор Microsoft (R) Visual C# версии 4.11.0-3.24460.3 (5649376e)
с Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\user\Documents\Visual Studio\OOP_Projects>MyGelloProgram.exe
Please enter your name
Plida
Hello, Plida

C:\Users\user\Documents\Visual Studio\OOP_Projects>ren MyGelloProgram.exe MyHelloProgram.exe
```

Рисунок 1.3 — Упр №1: Компиляция и вывод программы

## 2 УПРАЖНЕНИЕ №2. СОЗДАНИЕ ПРОГРАММЫ С ПОМОЩЬЮ СРЕДЫ РАЗРАБОТКИ VISUAL STUDIO .NET

Это упражнение нацелено на написание программы первого упражнения в среде разработки Visual Studio.NET.

В Visual Studio .NET было создано консольное приложение Greetings с решением Greeter. В нём, в файле Program.cs был вписан требуемый код запроса имени пользователя. Он представлен на рисунке 2.1.

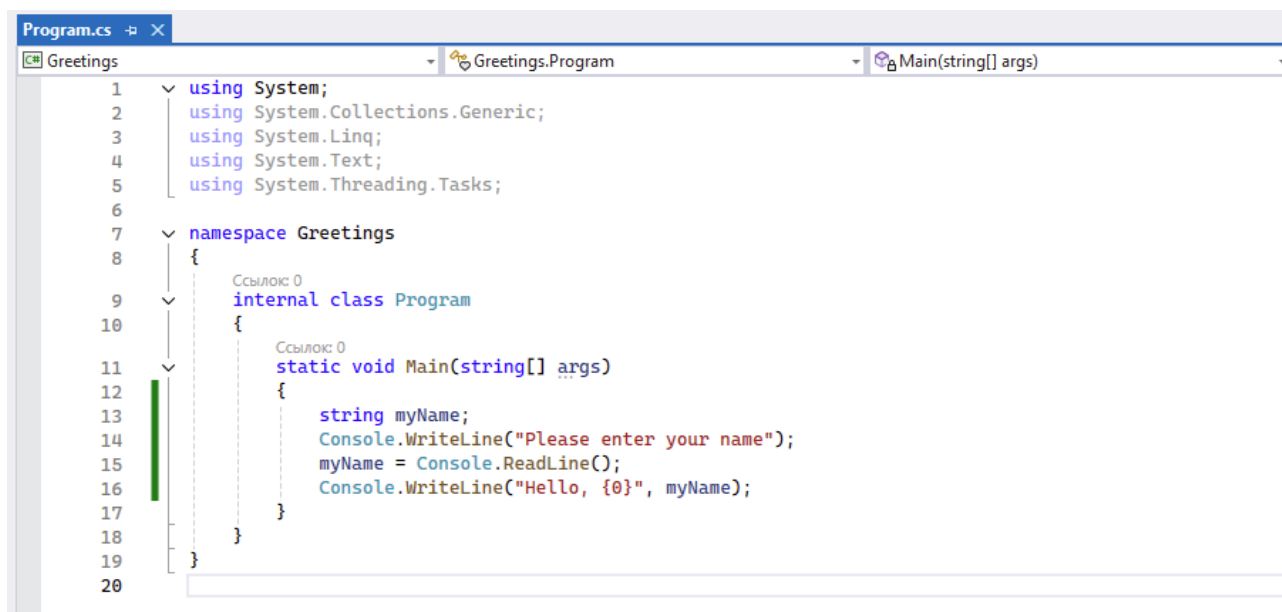


Рисунок 2.1 — Упр №2: Код программы

Процесс сборки программы можно увидеть на рисунке 2.1.

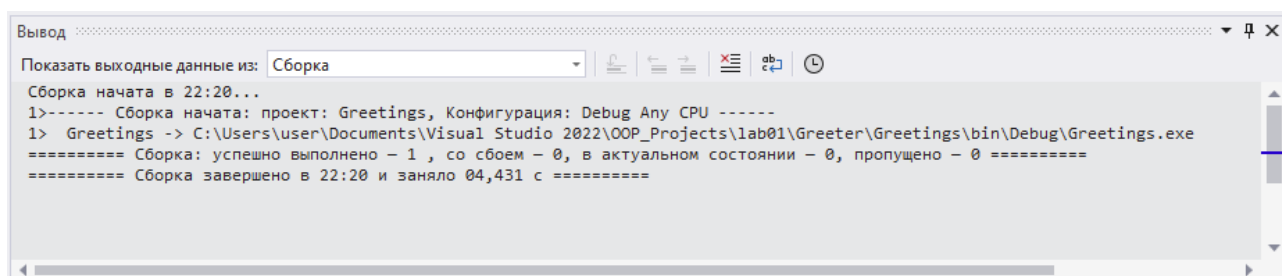


Рисунок 2.2 — Упр №2: Сборка программы

Затем полученная программа была запущена. Её вывод в консоли можно посмотреть на рисунке 2.3.

C:\Windows\system32\cmd.exe

```
Please enter your name  
Plida  
Hello, Plida  
Для продолжения нажмите любую клавишу . . .
```

Рисунок 2.3 — Упр №2: Запуск программы

### 3 УПРАЖНЕНИЕ №3. ИСПОЛЬЗОВАНИЕ ОТЛАДЧИКА VISUAL STUDIO .NET

Это задание направлено на обучение работы с интегрированным отладчиком Visual Studio .NET: прохождение программы по шагам и просматривания значений переменных.

В упражнении используется написанная в упражнении №2 программа приветствия пользователя.

Была поставлена точка остановке на строке «Console.WriteLine("Please enter your name")» и запущена отладка. В окно командной строки не вывелось ни одной строки: программа остановилась в указанной точке.

Затем во вкладке верхнего меню «Отладка» было открыто окно «Контрольные значения 1». Туда была добавлена пустая переменная «myName» типа string. Окно контрольных значений можно увидеть на рисунке 3.1.

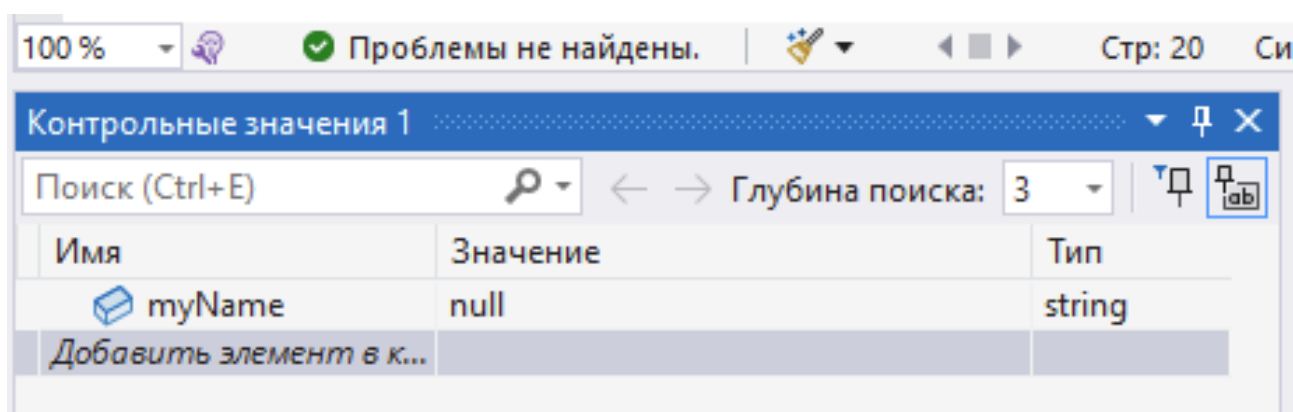


Рисунок 3.1 — Упр №3: Добавление переменной в контрольные значения

Был произведён один шаг с обходом и вывелась строка «Please enter your name».

Был совершён ещё один шаг, и программа запросила ввод имени, что было совершено. Имя «Plida» записалось в переменную «myName», что видно в контрольных значениях на рисунке 3.2.



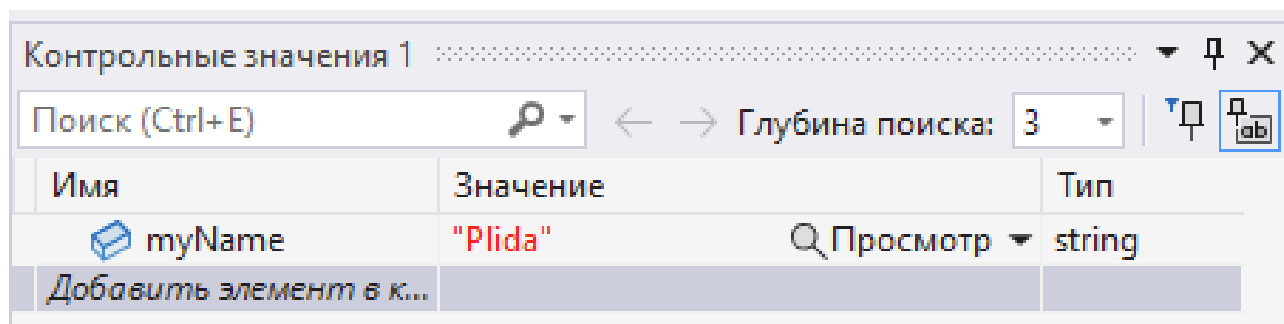


Рисунок 3.2 — Упр №3: Изменение переменной

Был произведён ещё один шаг с обходом, и программа вывела приветствие «Hello, Plida». Затем отладка была продолжена клавишей F5, чем программа была завершена.

## 4 УПРАЖНЕНИЕ №4. ДОБАВЛЕНИЕ В C#-ПРОГРАММУ ОБРАБОТЧИКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ

Это упражнение нацелено на написание программы, в которой используется обработчик исключительных ситуаций, отлавливающий ошибки времени выполнения. Программа должна запрашивать у пользователя два целых числа, делить первое число на второе и выводить полученный результат.

В Visual Studio .NET был создан новый проект Divider с решением DivideIt.

Была написана первая версия программы, чей код представлен на рисунке 4.1.

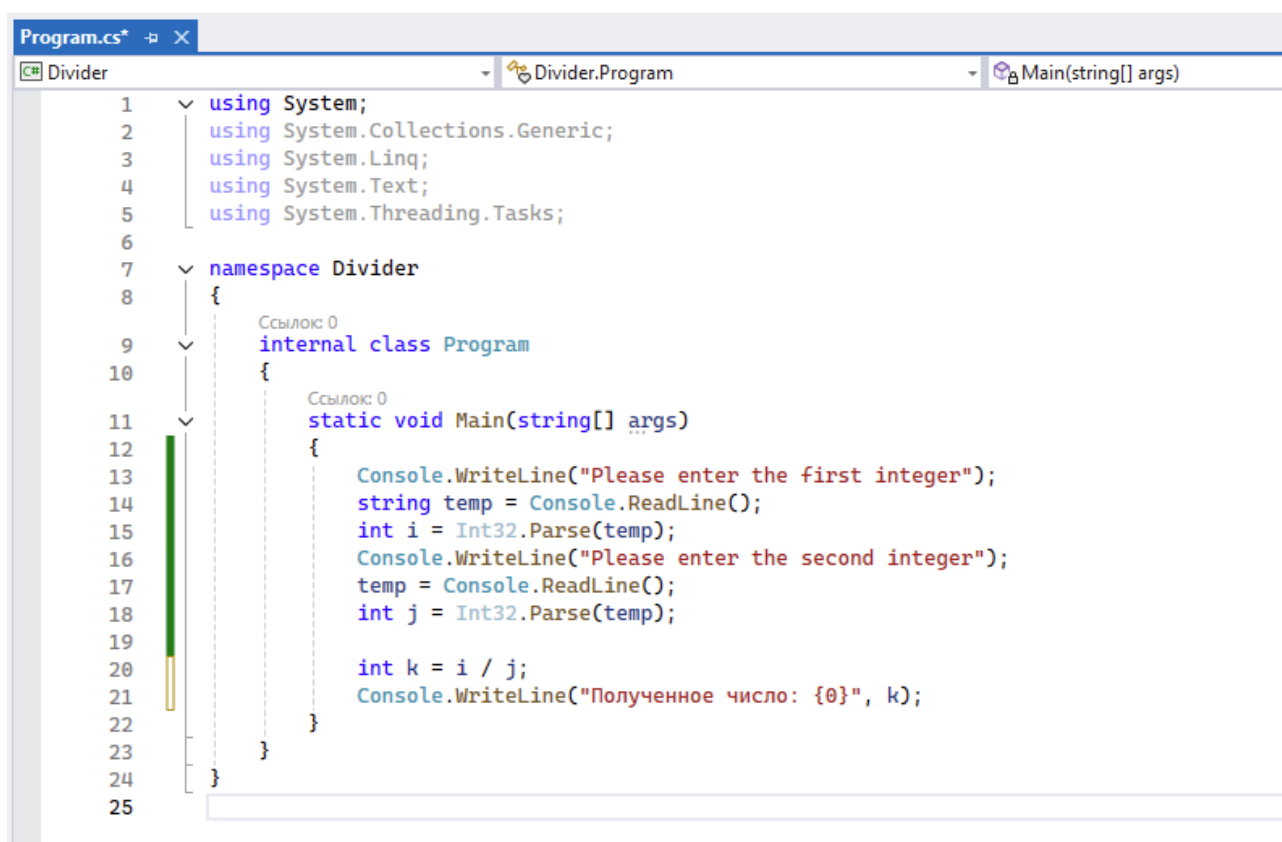
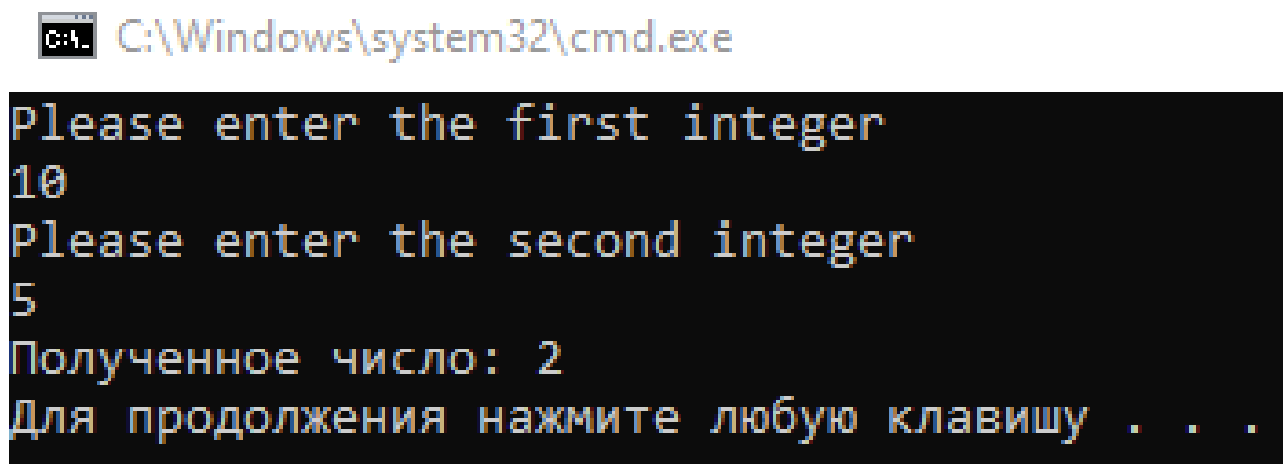


Рисунок 4.1 — Упр №4: Код программы без обработчика исключительных ситуаций

При введенных значениях «10» и «5» программа выдаёт корректное значение «2», что можно увидеть на рисунке 4.2.

При введенных значениях «10» и «3» программа бы выдала значение «3», так как она использует целочисленное деление, при котором значение округляется в меньшую сторону.



```
C:\Windows\system32\cmd.exe

Please enter the first integer
10
Please enter the second integer
5
Полученное число: 2
Для продолжения нажмите любую клавишу . . .
```

Рисунок 4.2 — Упр №4: Результат первой версии программы

Эта версия программы некорректно обрабатывает исключительную ситуацию деления на ноль и аварийно закрывается. На рисунке 4.3 показано окно Visual Studio .NET, сообщающее об ошибке.

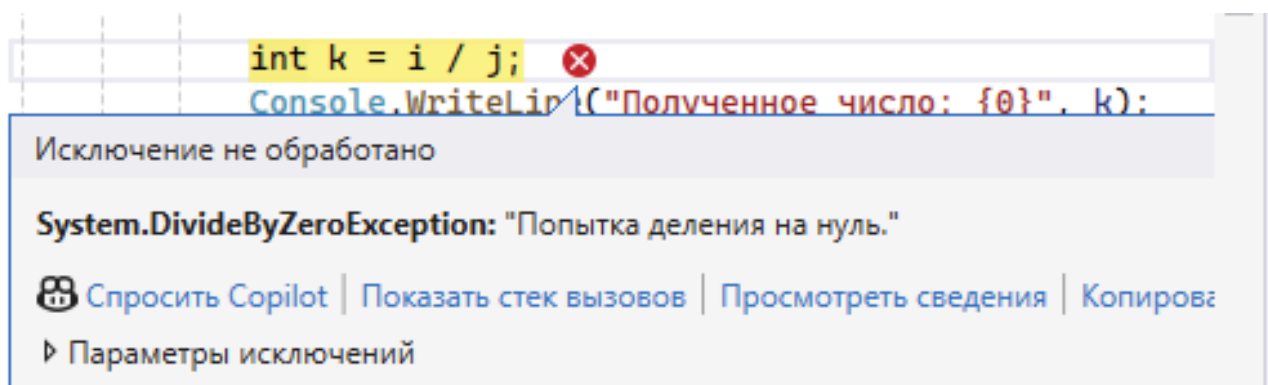


Рисунок 4.3 — Упр №4: Ошибка деления на ноль

В код программы был добавлен обработчик исключительных ситуаций через блоки try и catch. Версия программы с универсальным обработчиком показана на рисунке 4.4

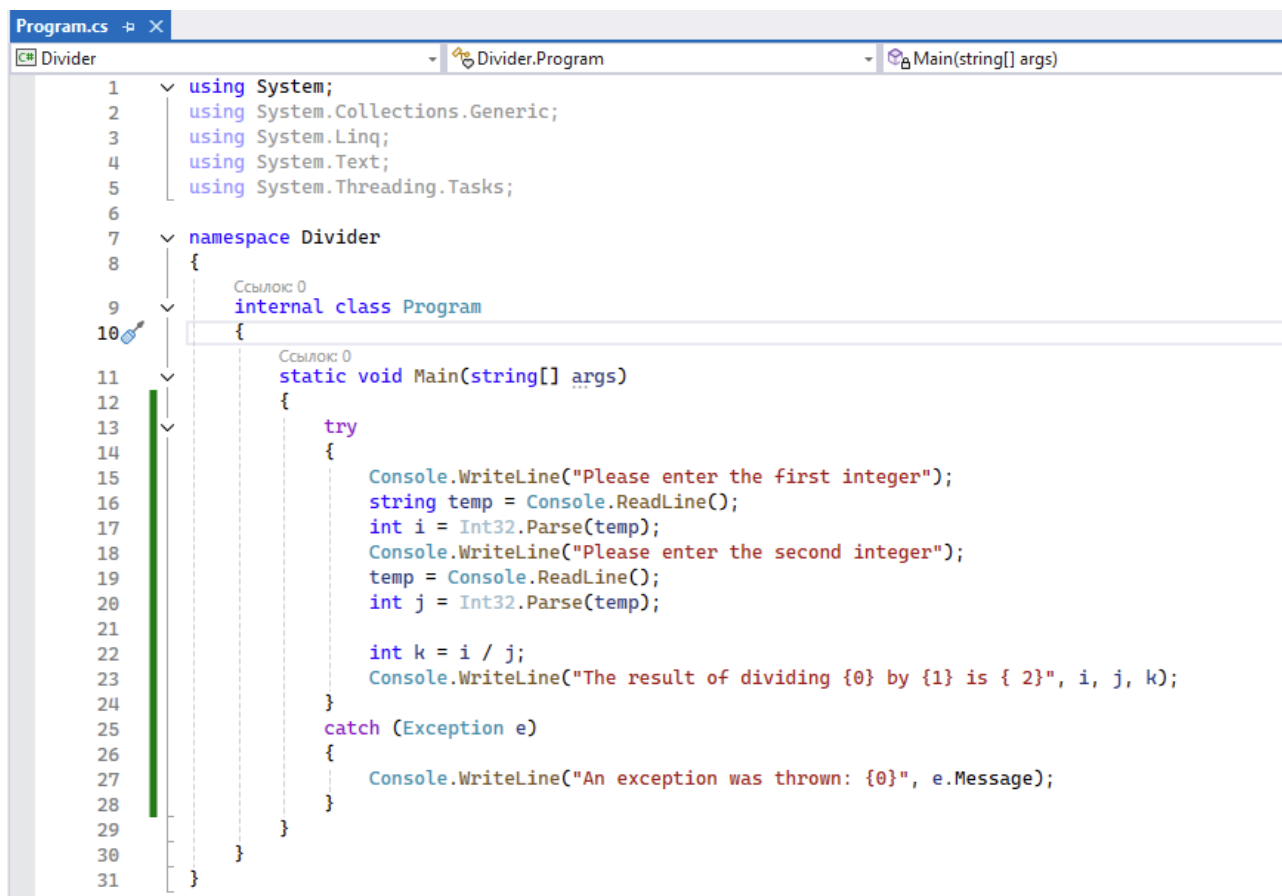


Рисунок 4.4 — Упр №4: Версия программы с универсальным обработчиком

При столкновении с исключительным случаем программа вместо аварийного закрытия выводит в консоль сообщение об ошибке.

Результат деления на ноль показан на рисунке 4.5.

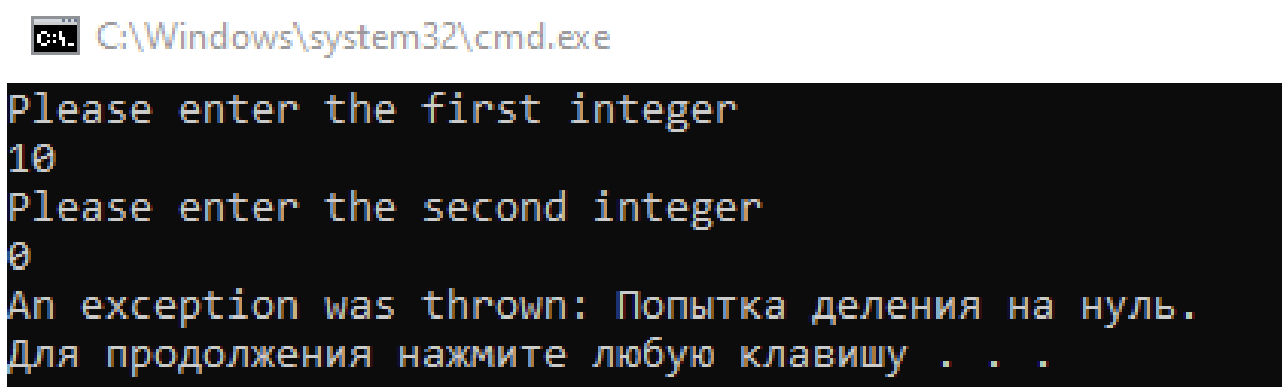


Рисунок 4.5 — Упр №4: Результат деления на ноль с обработчиком

Созданный универсальный обработчик способен ловить и другие исключительные случаи. При попытке ввести в программу не целое число, а букву,

выводится сообщение «An exception was thrown: Входная строка имела неверный формат.», и программа обрывается.

Был добавлен ещё один блок `catch` для случаев `FormatException` перед универсальным обработчиком. Данный обработчик выводит сообщение, показанное на рисунке 4.6.

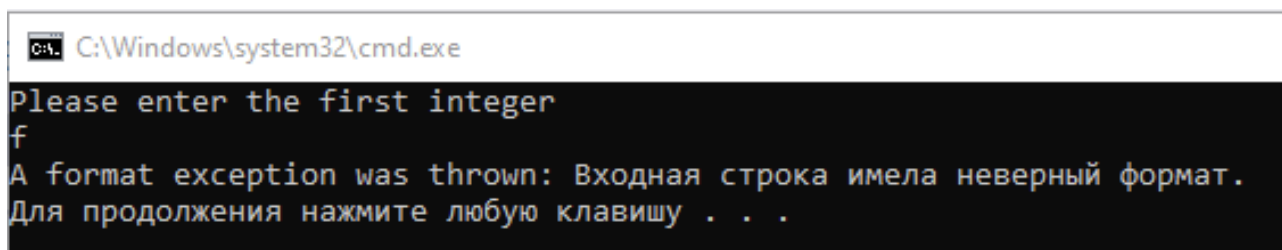


Рисунок 4.6 — Упр №4: Работа обработчика для `FormatException`

## 5 УПРАЖНЕНИЕ №5. РАСЧЁТ ПЛОЩАДИ ТРЕУГОЛЬНИКА

В данной программе требуется создать программу, подсчитывающего площадь равностороннего треугольника с заранее известным периметром. Программа должна брать пользовательское значение периметра, и выводить расчёты о длине стороны и площади на экран.

Был создан новый проект «shapes-areas» с решением «find-area».

В метод Main был добавлен обработчик исключительных ситуаций вида Exception и FormatException.

Программа принимает ввод пользователя через Console.Write(), записывая его в переменную P типа double. Она вычисляет полупериметр, деля P на два, и вычисляет сторону треугольника, деля P на три.

Площадь треугольника вычисляется по видоизменённой формуле Герона  $S = \sqrt{p(p - side)^3}$ . Метод вычисления квадратного корня взят из класса Math.

Написанный код программы представлен на рисунке 5.1

```
shapes-areas shapes_areas.Program Main(string[] args)
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace shapes_areas
8 {
9     Ссылка 0
10    internal class Program
11    {
12        Ссылка 0
13        static void Main(string[] args)
14        {
15            try
16            {
17                Console.WriteLine("Введите периметр треугольника: ");
18                double P = double.Parse(Console.ReadLine());
19                double side = P / 3;
20                double p = P / 2;
21                double area = Math.Sqrt(p * Math.Pow((p - side), 3.00));
22                Console.WriteLine($"Сторона Площадь");
23                Console.WriteLine("{0,-7:F2} {1,-7:F2}", side, area);
24            }
25            catch (FormatException e)
26            {
27                Console.WriteLine("An format exception was thrown: {0}", e.Message);
28            }
29            catch (Exception e)
30            {
31                Console.WriteLine("An exception was thrown: {0}", e.Message);
32            }
33        }
34    }
```

Рисунок 5.1 — Упр №5: Код программы

Пример выполнения программы представлен на рисунке 5.2.

```
C:\Windows\system32\cmd.exe
Введите периметр треугольника: 18
Сторона Площадь
6,00      15,59
Для продолжения нажмите любую клавишу . . .
```

Рисунок 5.2 — Упр №5: Вывод программы

## ЗАКЛЮЧЕНИЕ

При выполнении лабораторной работы были созданы несколько программ на языке C#. Были изучены методы компиляции как через консольную строку, так и через среду разработки Visual Studio .NET. Был изучен отладчик Visual Studio .NET и добавлен обработчик исключительных ситуаций. Наконец, была создана программа вычисления площади равностороннего треугольника по введённому периметру.

Цель изучения структуры программы на языке C# и приобретение навыков ее компиляции и отладки была выполнена.