

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет инфокоммуникационных технологий

Отчет по дисциплине: **«Объектно-ориентированное программирование»**
Лабораторная работа 1. "Создание программы спомощью среды разработки
Visual Studio.NET"

Выполнил: Абдулов
Илья Александрович

Группа: К3221

Проверил: Васильев
Сергей Юрьевич

Санкт-Петербург
2023

Содержание

| | |
|--|----|
| Введение | 3 |
| 1 Создание программы на C# | 4 |
| 1.1 Упражнение 1. Создание простой программы в тексто- вом редакторе | 4 |
| 1.2 Упражнение 2. Создание программы с помощью среды разработки Visual Studio | 6 |
| 1.3 Упражнение 3. Использование отладчика Visual Studio . | 7 |
| 1.4 Упражнение 4. Добавление в C#-программу обработчи- ка исключительных ситуаций | 9 |
| 1.5 Упражнение 5. Расчет площади треугольника | 11 |
| Заключение | 12 |

Введение

Целью работы является изучение структуры программы на языке C# и приобретение навыков ее компиляции и отладки. Для достижения поставленной цели необходимо решить следующие упражнения:

1. Создать простую программу на C# в текстовом редакторе;
2. Создать программу с помощью среды разработки Visual Studio;
3. Использовать отладчик Visual Studio;
4. Добавить в C#-программу обработчика исключительных ситуаций;
5. Рассчитать площадь треугольника в программе.

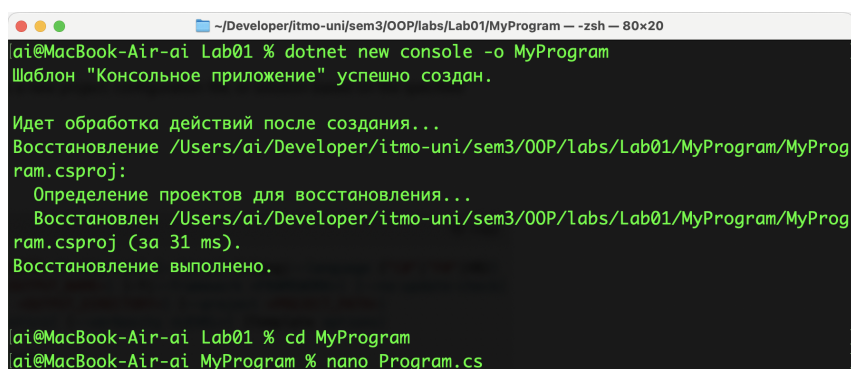
Лабораторная выполняется на компьютере под macOS, поэтому некоторые этапы и инструменты выполнения работы будут отличаться от рекомендуемых в Windows.

1 Создание программы на C#

В данном разделе изучается структура программы на языке C#, компиляция и отладка программ.

1.1 Упражнение 1. Создание простой программы в текстовом редакторе

Откроем окно терминала на macOS и создадим новый консольный проект dotnet. Откроем текстовый редактор nano и изменим первую программу на C# (Рисунок 1.1).



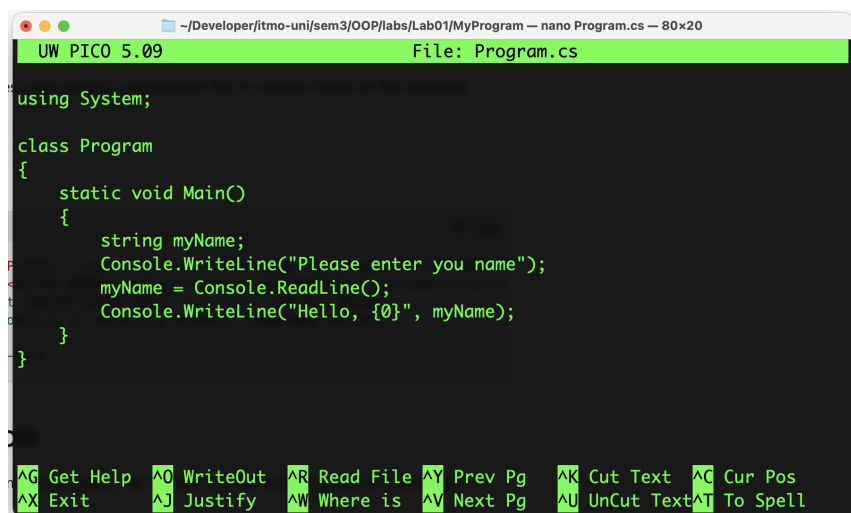
```
~/Developer/itmo-uni/sem3/OOP/labs/Lab01/MyProgram -- zsh -- 80x20
ai@MacBook-Air-ai Lab01 % dotnet new console -o MyProgram
Шаблон "Консольное приложение" успешно создан.

Идет обработка действий после создания...
Восстановление /Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/MyProgram/MyProgram.csproj:
  Определение проектов для восстановления...
  Восстановлен /Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/MyProgram/MyProgram.csproj (за 31 ms).
  Восстановление выполнено.

ai@MacBook-Air-ai Lab01 % cd MyProgram
ai@MacBook-Air-ai MyProgram % nano Program.cs
```

Рисунок 1.1 — Первая программа на C#

Определим пространство имен System, создадим класс с именем Program, добавим в этот класс метод Main (Рисунок 1.2). В методе Main создадим переменную myName типа string, попросим пользователя ее ввести и после выведем имя пользователя.

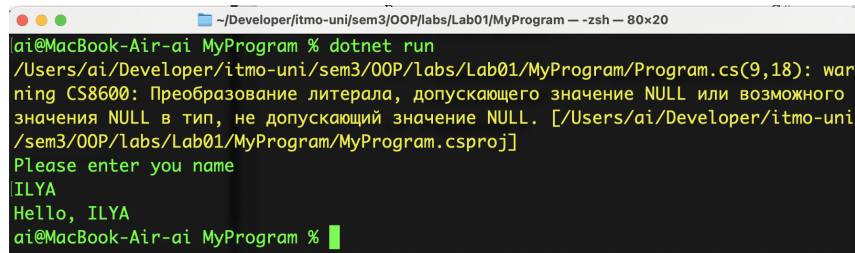


```
UW PICO 5.09 File: Program.cs
using System;

class Program
{
    static void Main()
    {
        string myName;
        Console.WriteLine("Please enter you name");
        myName = Console.ReadLine();
        Console.WriteLine("Hello, {0}", myName);
    }
}
```

Рисунок 1.2 — Код программы на C#

Запустим программу в командной строке, введем имя и посмотрим вывод (Рисунок 1.3).



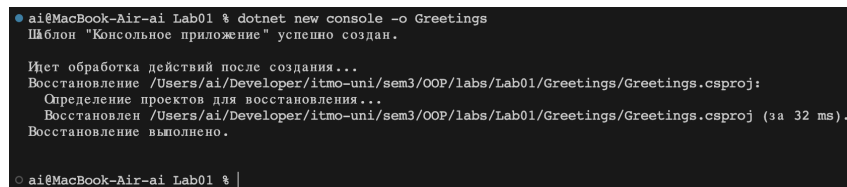
```
ai@MacBook-Air-ai MyProgram % dotnet run
/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/MyProgram/Program.cs(9,18): warning CS8600: Преобразование литерала, допускающего значение NULL или возможного значения NULL в тип, не допускающий значение NULL. [/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/MyProgram/MyProgram.csproj]
Please enter you name
ILYA
Hello, ILYA
ai@MacBook-Air-ai MyProgram %
```

Рисунок 1.3 — Вывод программы на C#

1.2 Упражнение 2. Создание программы с помощью среды разработки Visual Studio

В этом упражнении мы перепишем программу первого упражнения но воспользуемся уже редактором Visual Studio Code.

Создадим новый проект C# с названием Greetings в директории Lab01 (Рисунок 1.4). Перепишем код из прошлого упражнения, но изменим название класса на Greeter (Рисунок 1.5).

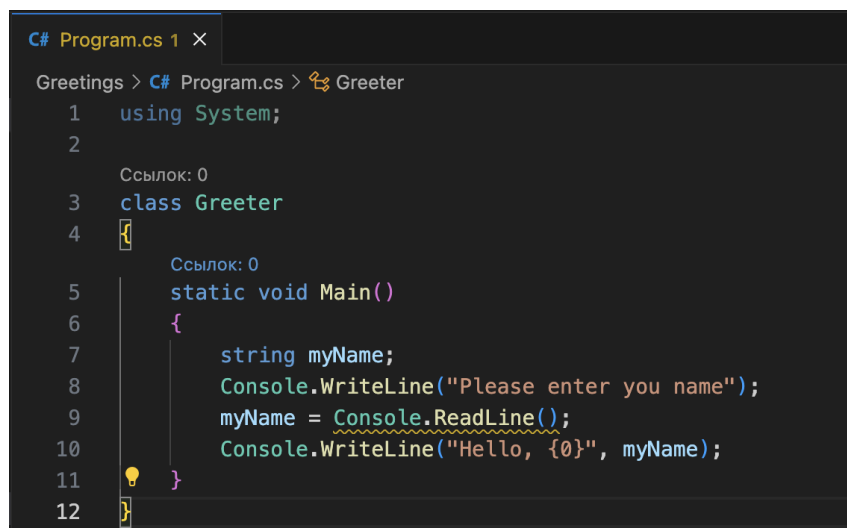


```
ai@MacBook-Air-ai Lab01 % dotnet new console -o Greetings
Шаблон "Консольное приложение" успешно создан.

Идет обработка действий после создания...
Восстановление /Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Greetings/Greetings.csproj:
Определение проектов для восстановления...
Восстановлен /Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Greetings/Greetings.csproj (за 32 ms).
Восстановление выполнено.

ai@MacBook-Air-ai Lab01 %
```

Рисунок 1.4 — Создание проекта Greetings



```
C# Program.cs 1 x
Greetings > C# Program.cs > Greeter
1  using System;
2
   Ссылка: 0
3  class Greeter
4  {
   Ссылка: 0
5      static void Main()
6      {
7          string myName;
8          Console.WriteLine("Please enter you name");
9          myName = Console.ReadLine();
10         Console.WriteLine("Hello, {0}", myName);
11     }
12 }
```

Рисунок 1.5 — Код класса Greeter

Запустим новое приложение и убедимся, что все работает корректно.

1.3 Упражнение 3. Использование отладчика Visual Studio

В этом задании научимся работать с интегрированным отладчиком Visual Studio Code: проходить программу по шагам и просматривать значения переменных.

Откроем проект Greetings и на уровне строки кода, где впервые встречается `Console.WriteLine` поставим красную точку слева, означающую остановку отладки.

Запустим C# Debugger и увидим, что дебаггер остановится в месте точки остановки. При этом слева в меню переменных будет видно, что переменная `myName` типа `string` равна `null` на этом этапе выполнения программы (Рисунок 1.6).

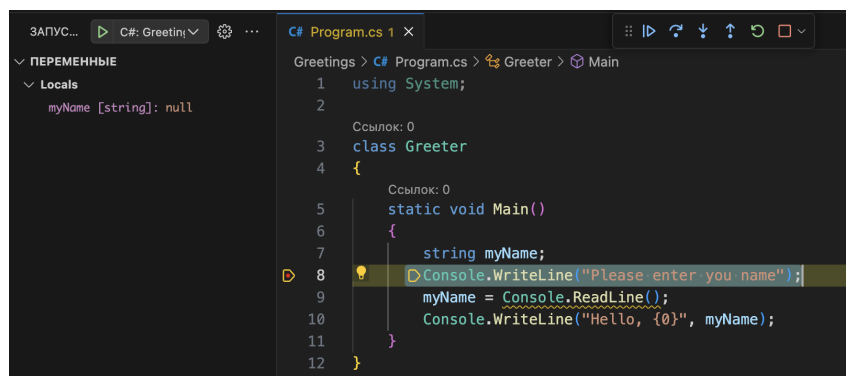


Рисунок 1.6 — Значение переменной `myName` при отладке

Используем команды пошагового выполнения и увидим, что после 9 строчки кода измениться значение переменной `myName` на введенное в терминале (Рисунок 1.7).

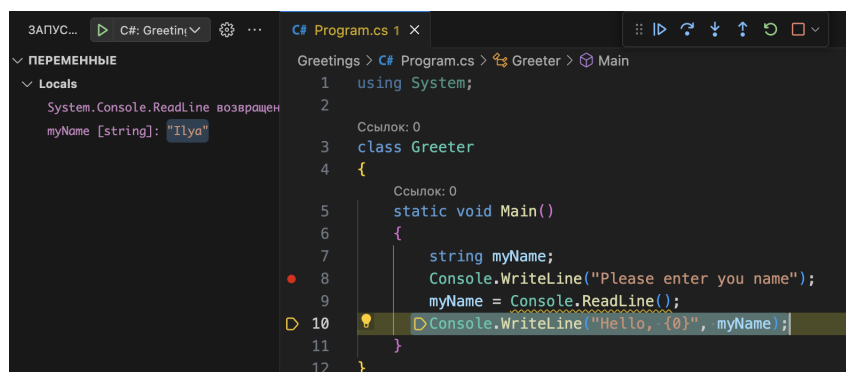
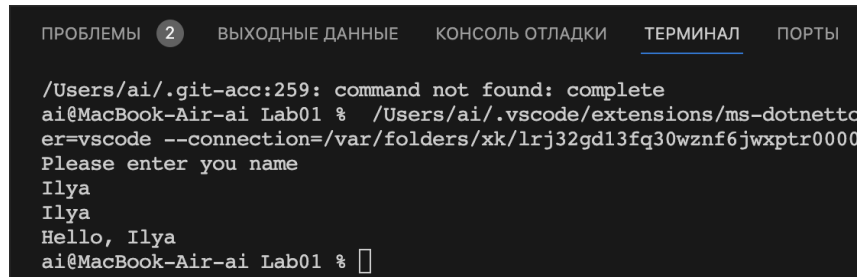


Рисунок 1.7 — Новое значение `myName`

После завершения отладки в консоли будет выведено приветствие (Рисунок 1.8).



```
ПРОБЛЕМЫ 2 Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

/Users/ai/.git-acc:259: command not found: complete
ai@MacBook-Air-ai Lab01 % /Users/ai/.vscode/extensions/ms-dotnetto
er=vscode --connection=/var/folders/xk/lrj32gd13fq30wznf6jwxptr0000
Please enter you name
Ilya
Ilya
Hello, Ilya
ai@MacBook-Air-ai Lab01 %
```

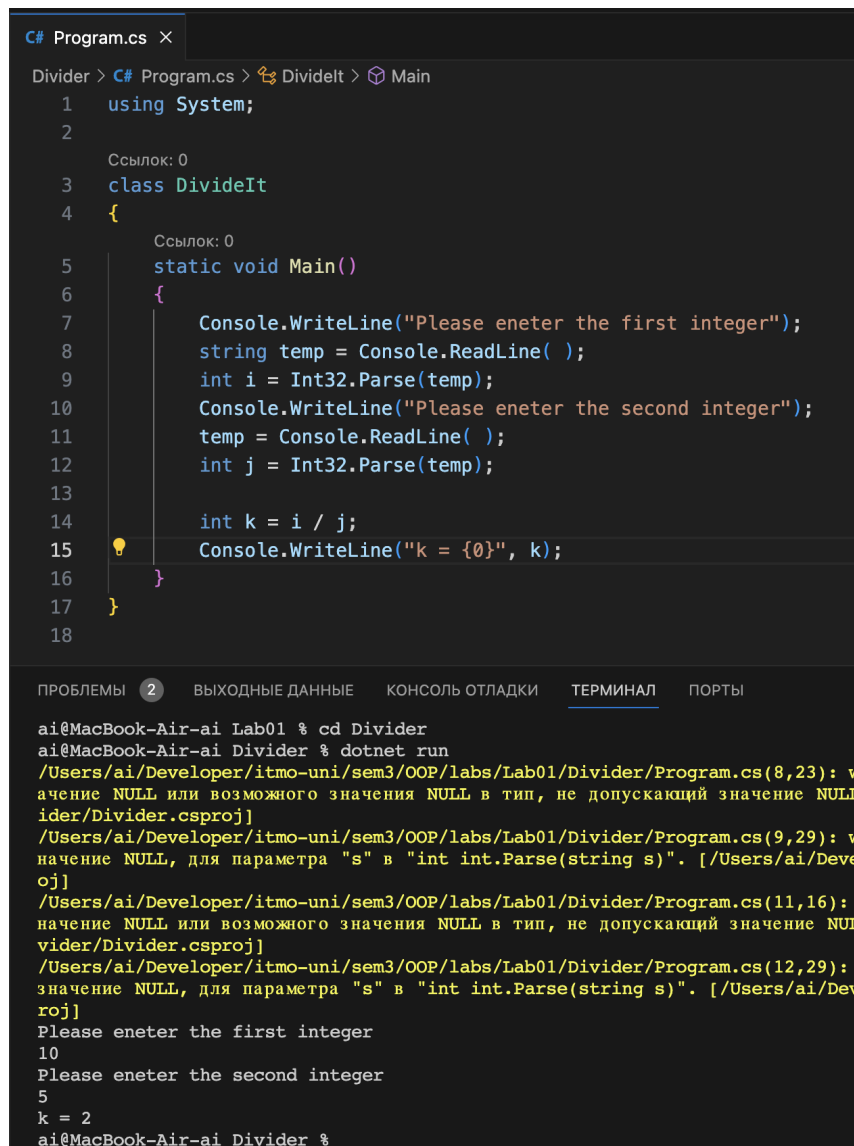
Рисунок 1.8 — Вывод консоли

1.4 Упражнение 4. Добавление в C#-программу обработчика исключительных ситуаций

В этом упражнении будет написана программа с обработчиком исключительных ситуаций, который будет отлавливать ошибки времени выполнения.

Создадим новый проект Divider и в нем класс DivideIt.

В методе Main класса DivideIt напомним код, запрашивающий у пользователя два числа. Преобразуем полученные числа в тип `int`. Создадим новую переменную типа `int`, в которой будет храниться результат деления чисел. Протестируем программу на примере ввода чисел 10 и 5 (Рисунок 1.9).



```
C# Program.cs x
Divider > C# Program.cs > DivideIt > Main
1  using System;
2
3  Ссылка: 0
   class DivideIt
4  {
5      Ссылка: 0
       static void Main()
6      {
7          Console.WriteLine("Please eneter the first integer");
8          string temp = Console.ReadLine( );
9          int i = Int32.Parse(temp);
10         Console.WriteLine("Please eneter the second integer");
11         temp = Console.ReadLine( );
12         int j = Int32.Parse(temp);
13
14         int k = i / j;
15         Console.WriteLine("k = {0}", k);
16     }
17 }
18

ПРОБЛЕМЫ 2 Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

ai@MacBook-Air-ai Lab01 % cd Divider
ai@MacBook-Air-ai Divider % dotnet run
/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Divider/Program.cs(8,23): warning CS8632: The type 'int' is not a valid type for the 'Parse' method. [/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Divider/Divider.csproj]
/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Divider/Program.cs(9,29): warning CS8632: The type 'int' is not a valid type for the 'Parse' method. [/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Divider/Divider.csproj]
/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Divider/Program.cs(11,16): warning CS8632: The type 'int' is not a valid type for the 'Parse' method. [/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Divider/Divider.csproj]
/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Divider/Program.cs(12,29): warning CS8632: The type 'int' is not a valid type for the 'Parse' method. [/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/Divider/Divider.csproj]
Please eneter the first integer
10
Please eneter the second integer
5
k = 2
ai@MacBook-Air-ai Divider %
```

Рисунок 1.9 — Программа деления двух чисел

Протестируем программу на примере других чисел и увидим, что в программе возникает исключительная ситуация, когда второе число 0.

Добавим в программу обработчик исключительных ситуаций, используя блок `try` и несколько блоков `catch`.

Протестируем код обработчика исключений на случаях, когда второе число равно 0 (Рисунок 1.10) и когда пользователем введены буквы вместо чисел (Рисунок 1.11).

```
Please eneter the first integer
10
Please eneter the second integer
0
An exception was thrown: Attempted to divide by zero.
ai@MacBook-Air-ai Divider % |
```

Рисунок 1.10 — Вывод обработчика исключений при вводе 0

```
ai@MacBook-Air-ai Divider % dotnet run
Please eneter the first integer
8394
Please eneter the second integer
abacaba
A format exception was thrown: The input string 'abacaba' was not in a correct format.
ai@MacBook-Air-ai Divider %
```

Рисунок 1.11 — Вывод при вводе букв

На этот раз ошибка перехватывается и на экран выводится соответствующее сообщение.

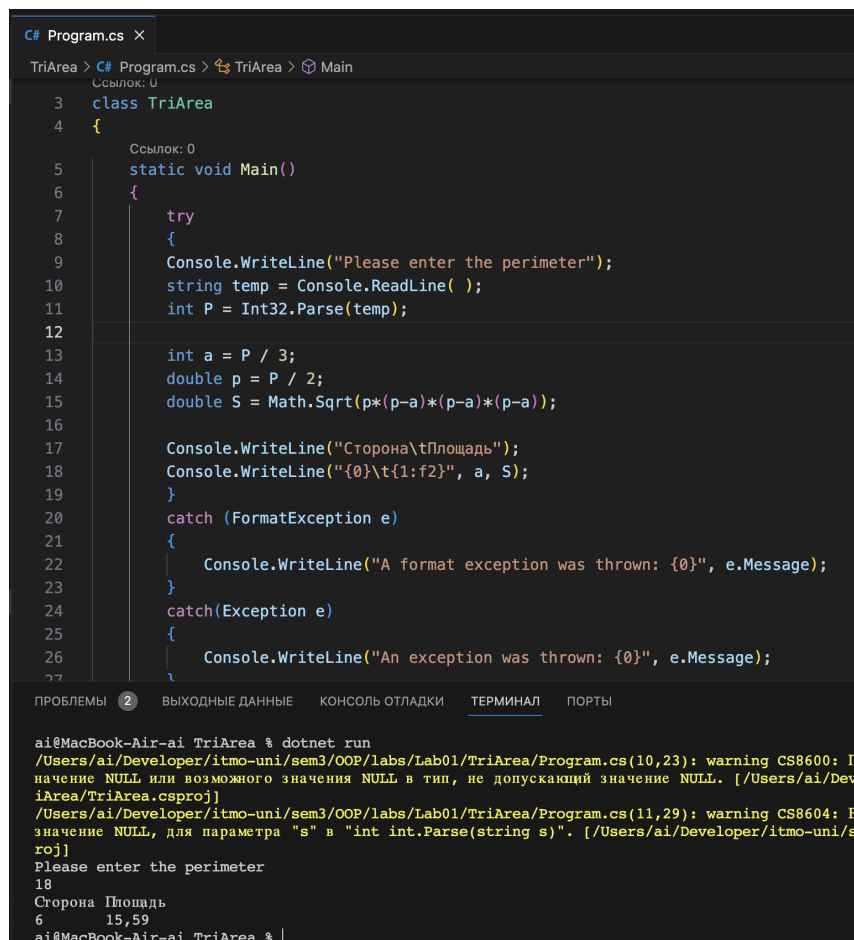
1.5 Упражнение 5. Расчет площади треугольника

Создадим программу, подсчитывающую площадь равностороннего треугольника, периметр которого известен.

Пользователь вводит значение периметра. Расчитаем сторону треугольника и площадь по формуле Герона.

При выводе будем использовать параметры форматирования C#.

Представим соответствующий код и вывод программы (Рисунок 1.12).



```
C# Program.cs X
TriArea > C# Program.cs > TriArea > Main
Ссылка: 0
3 class TriArea
4 {
5     Ссылка: 0
6     static void Main()
7     {
8         try
9         {
10            Console.WriteLine("Please enter the perimeter");
11            string temp = Console.ReadLine();
12            int P = Int32.Parse(temp);
13
14            int a = P / 3;
15            double p = P / 2;
16            double S = Math.Sqrt(p*(p-a)*(p-a));
17
18            Console.WriteLine("Сторона\tПлощадь");
19            Console.WriteLine("{0}\t{1:f2}", a, S);
20        }
21        catch (FormatException e)
22        {
23            Console.WriteLine("A format exception was thrown: {0}", e.Message);
24        }
25        catch (Exception e)
26        {
27            Console.WriteLine("An exception was thrown: {0}", e.Message);
28        }
29    }
30 }

ПРОБЛЕМЫ 2 ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

ai@MacBook-Air-ai TriArea % dotnet run
/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/TriArea/Program.cs(10,23): warning CS8600: Р
начение NULL или возможного значения NULL в тип, не допускающий значение NULL. [/Users/ai/Dev
iArea/TriArea.csproj]
/Users/ai/Developer/itmo-uni/sem3/OOP/labs/Lab01/TriArea/Program.cs(11,29): warning CS8604: Р
значение NULL, для параметра "s" в "int int.Parse(string s)". [/Users/ai/Developer/itmo-uni/s
roj]
Please enter the perimeter
18
Сторона Площадь
6 15,59
ai@MacBook-Air-ai TriArea %
```

Рисунок 1.12 — Код и вывод программы расчета площади

Заключение

В результате проделанной лабораторной работы была изучена структура программы на языке C# и приобретены навыки ее компиляции и отладки. Были решены следующие задачи:

1. Создана простая программа на C# в текстовом редакторе;
2. Создана программа с помощью среды разработки Visual Studio;
3. Использован отладчик Visual Studio;
4. Добавлен в C#-программу обработчик исключительных ситуаций;
5. Расчитана площадь треугольника в программе.