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

Факультет Инфокоммуникационных Технологий

**Лабораторная работа №1**

Выполнил:

Зенин Д.Д.

Проверил Иванов С. Е.

Санкт-Петербург,

2024

Оглавление

[ВВЕДЕНИЕ 3](#_Toc176866944)

[1. Упражнение 1 4](#_Toc176866945)

[2. Упражнение 2 4](#_Toc176866948)

[3. Упражнение 3 5](#_Toc176866950)

[4. Упражнение 4 7](#_Toc176866952)

[5. Упражнение 5 10](#_Toc176866953)

[ЗАКЛЮЧЕНИЕ 12](#_Toc176866954)

# ВВЕДЕНИЕ

Цель лабораторной работы №1 состояла в изучении структуры программы на языке С# и приобретении навыков ее компиляции и отладки.

# Упражнение 1

# В этом упражнении создам программу на языке С#, используя блокнот. Программа будет спрашивать имя и выводить приветственное сообщение.

# 

Рисунок 1 – Программа на языке С#, выводящая приветственное сообщение

Далее зайдем в Visual Studio и в командной строке проведем компиляцию программы. После успешной компиляции запустим ее и получим результат:

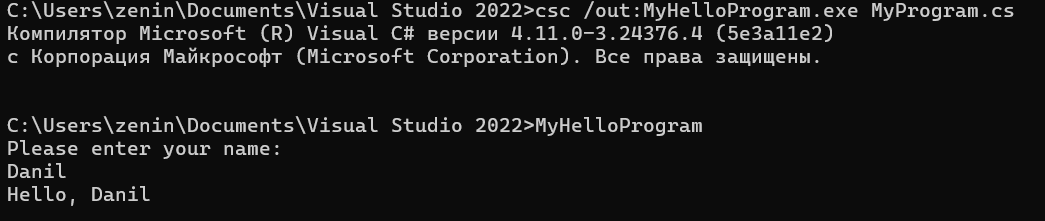


Рисунок 2 – Компиляция программы и ее запуск

Компилятор ошибок не выдает и программа работает.

# Упражнение 2

# Во втором упражнении необходимо было запустить и написать программу в среде разработки Visual Studio. Для этого создал новый проект(консольное приложение) и вставил туда код из прошлого упражнения:

Рисунок 3 – Код приложения в Visual Studio

После необходимо выполнить сборку проекта и запустить без отладки. Компилятор ошибок не выдает и все работает.

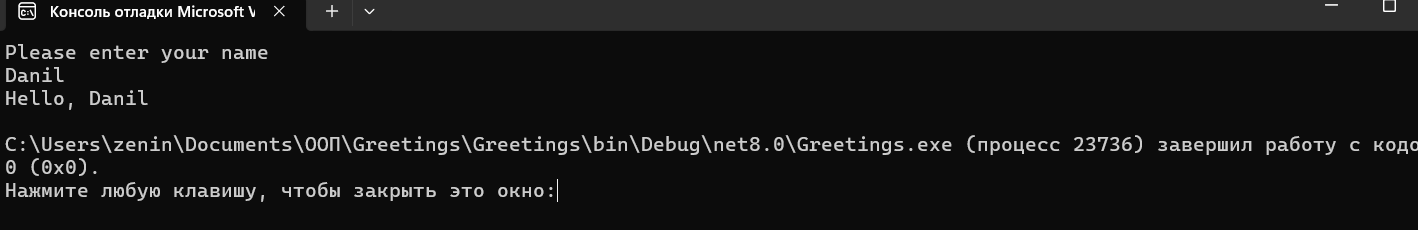


Рисунок 4 – Результат работы программы

Упражнение выполнено.

# Упражнение 3

# В данном упражнении необходимо поработать с приложением в режиме отладки. Для этого необходимо выставить на необходимой строке точку Останова и запустить режим отладки:

Рисунок 5 – Запуск программы в режиме отладки

В режиме отладки можно наблюдать текущее значение полей программы шаг за шагом после точки Останова.

После ввода имени мы можем наблюдать, что значение поля myName сменилось с null на Danil. Режим отладки работает.

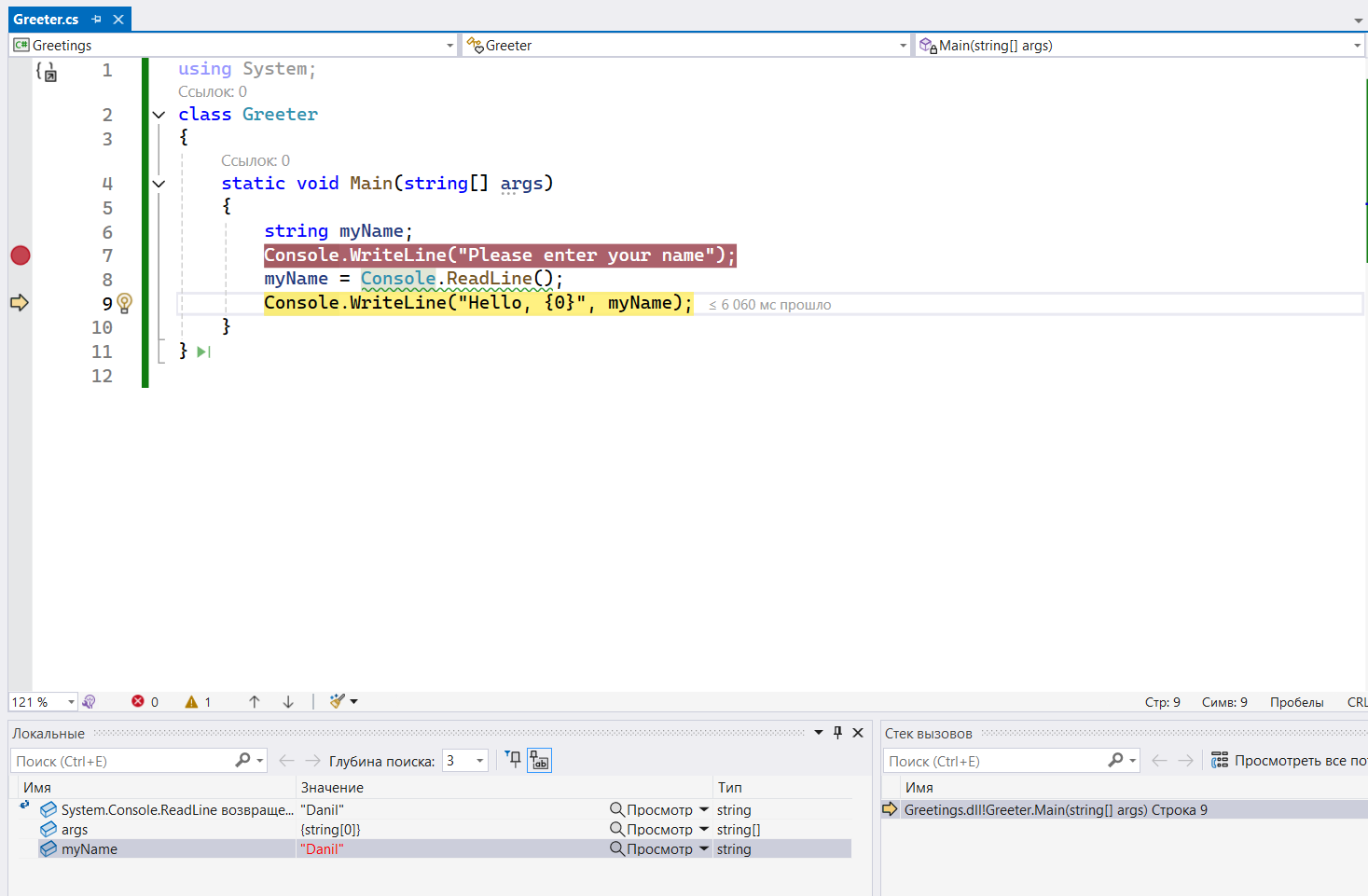


Рисунок 6 – Обновление значения поля в режиме отладки

# Упражнение 4

В упражнении 4 напишем приложение и добавим в него обработку исключений. Для этого создадим новый проект и добавим в него следующий код:

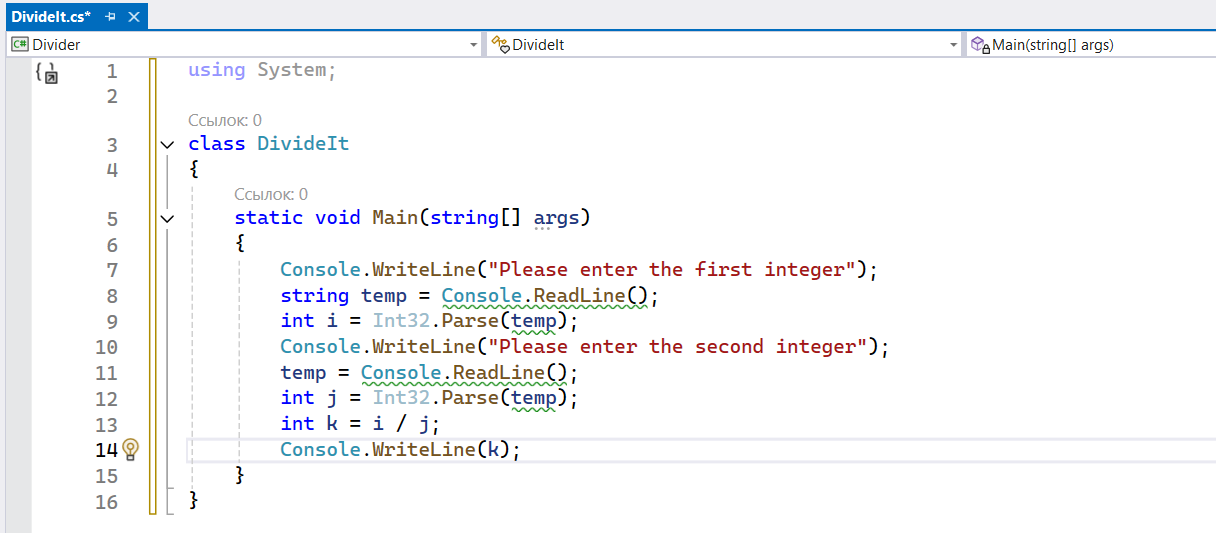
****

Рисунок 7 – Код приложения

Программа работает, и компилятор ошибок не выдает

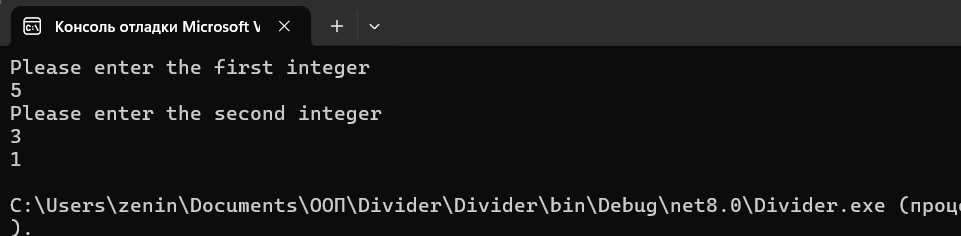


Рисунок 8 – Результат работы программы

Но если вторым числом ввести ноль, то выдаст ошибку:

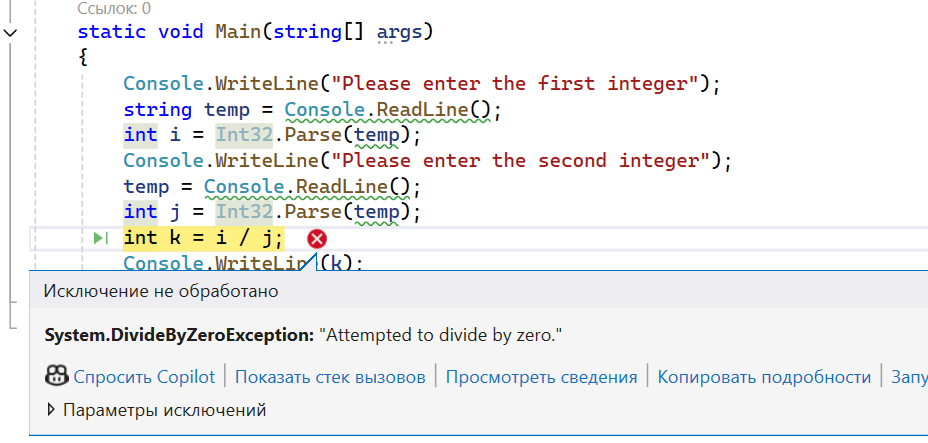


Рисунок 9 – Ошибка при делении на ноль

Для исправления теоретических ошибок программы необходимо создание исключений. Создадим обработчик исключений на неверный формат и на все остальные, чтобы точно не допустить ошибок.

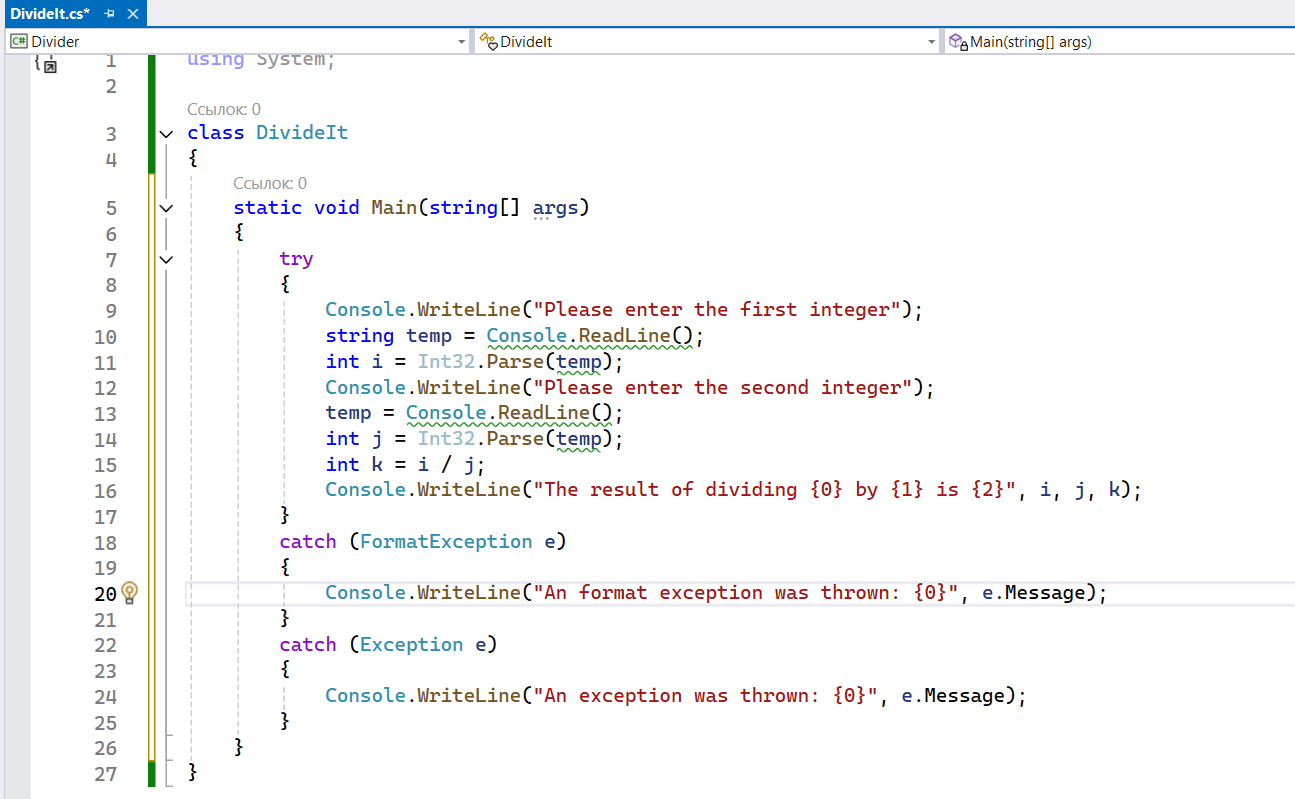


Рисунок 10 – Код программы с обработчиками исключений

Теперь, если ввести число, то будет такой вывод и не будет ошибок компилятора:

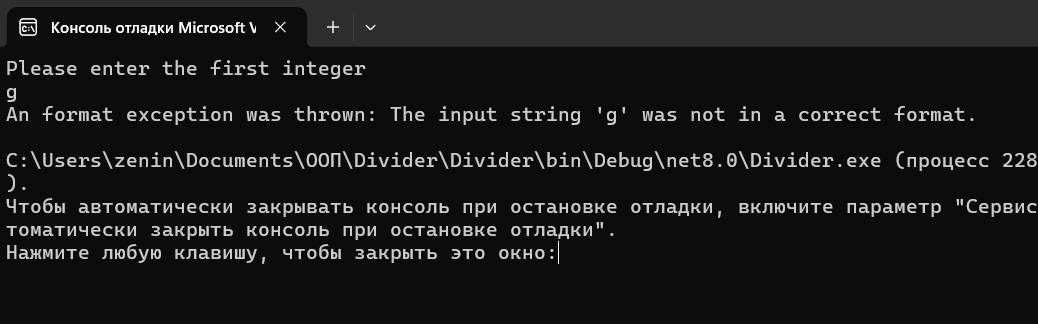


Рисунок 11 – Вывод программы с вводом ошибочного формата

А если поделить на ноль, то выведется ошибка деления на ноль:

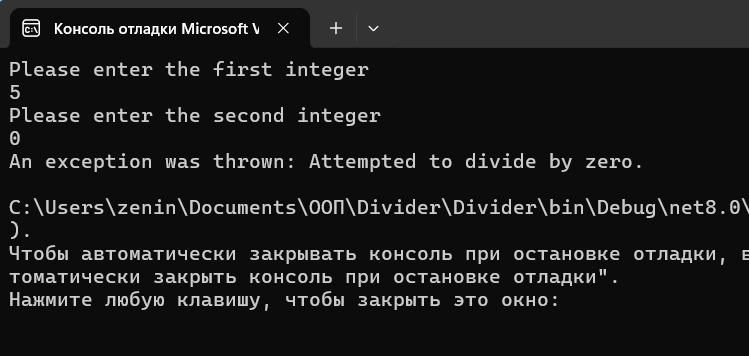


Рисунок 12 – Вывод программы с вводом деления на ноль

# Упражнение 5

В упражнении нужно написать приложение, которое вычисляет площадь равностороннего треугольника по его периметру. Использовал формулу Герона и метод sqrt() модуля Math

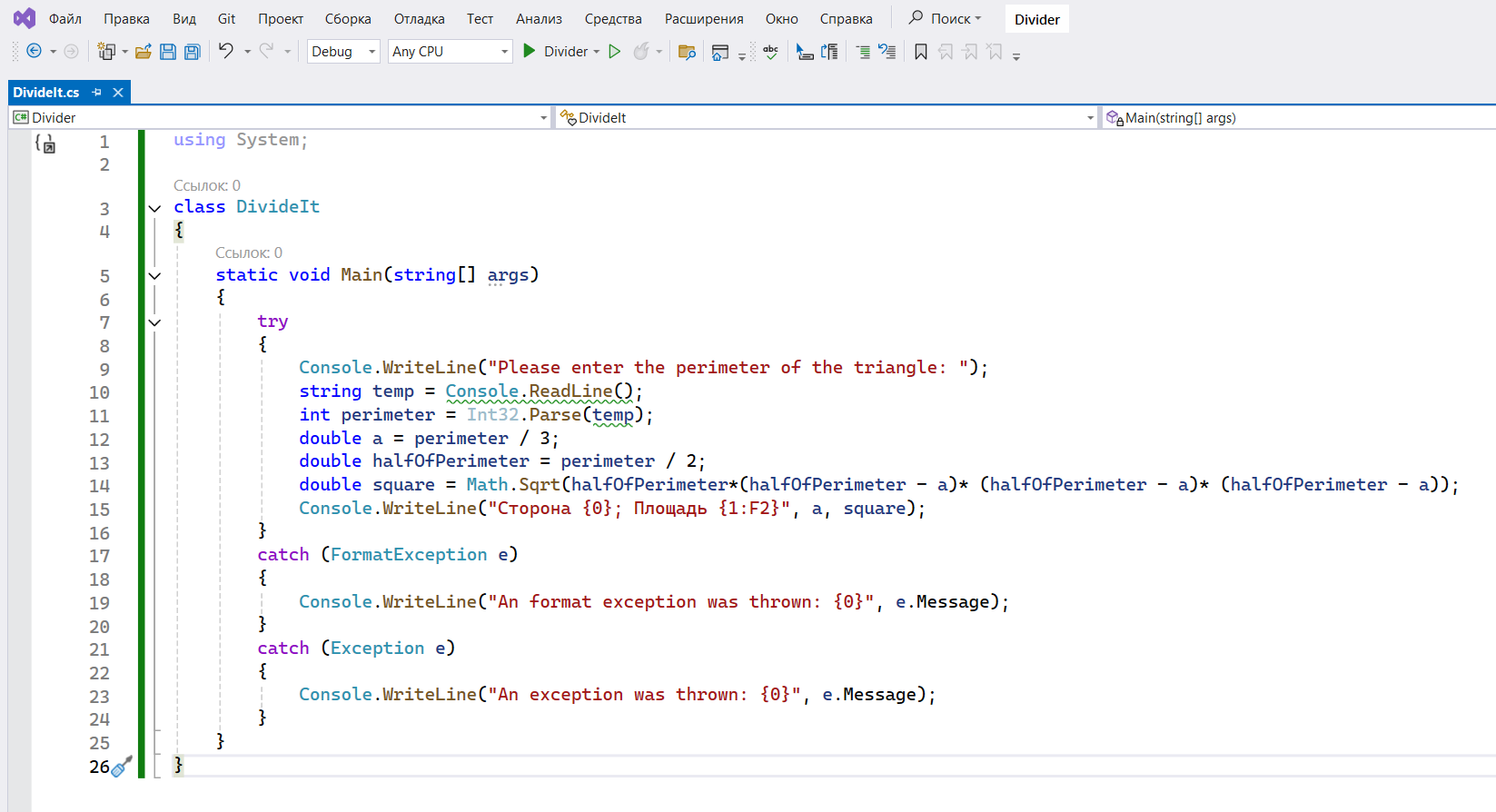


Рисунок 13 – Код программы

Программы обрабатывает возможные ошибки и выводит на экран ответ

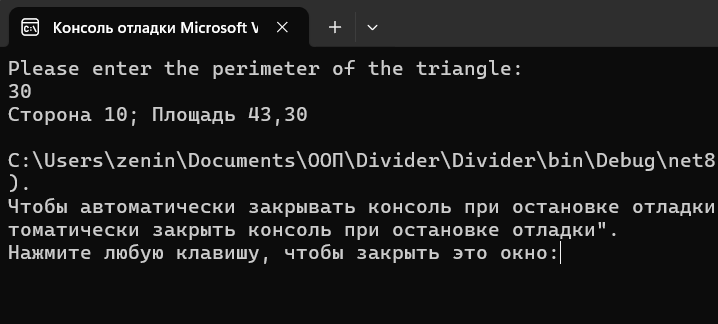


Рисунок 14 – Вывод программы

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы №1 были получены навыки работы с языков программирования С#