

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра информационной безопасности**

**ОТЧЕТ**  
**по лабораторной работе № 1**  
**по дисциплине «Объектно-ориентированное программирование»**  
**Тема: Обзор языка C#**

Студент гр. 3363

Преподаватель

_____	Минко Д.А., Овсейчик Н.И.
_____	Новакова Н.Е.

Санкт-Петербург  
2024

## **Цель работы**

Целью данной лабораторной работы является освоение основ программирования на языке C# с использованием среды разработки Microsoft Visual Studio 2022. В рамках необходимо научиться создавать простые консольные приложения, компилировать и запускать их как через Visual Studio, так и через командную строку, использовать встроенные инструменты отладки, а также реализовывать обработку исключений для обеспечения надежности программ.

## Ход работы

### 1. Создание простой программы на C# (рис. 1).

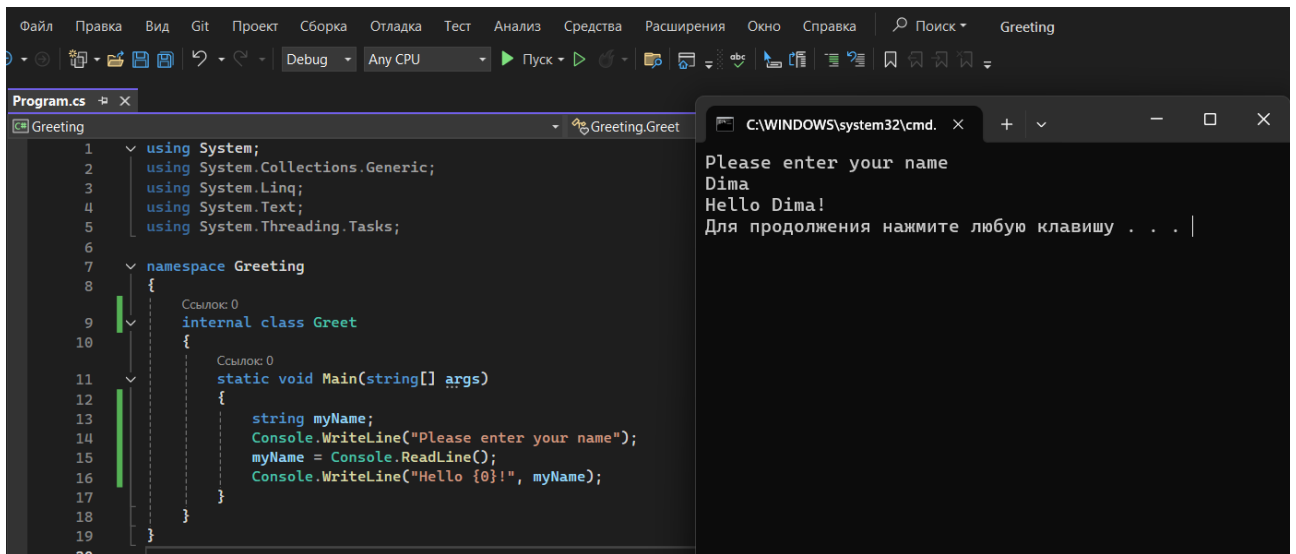


Рисунок 1 – Программа с запросом данных и их выводом в консоль

### 2. Компиляция и запуск программы C# из командной строки (рис. 2).

Из-за того, что версия Visual Studio 2022, её компилятор не поддерживает C# до 5-ой версии, следовательно запустить программу через консоль не предоставляется возможным, без дополнительного материала.

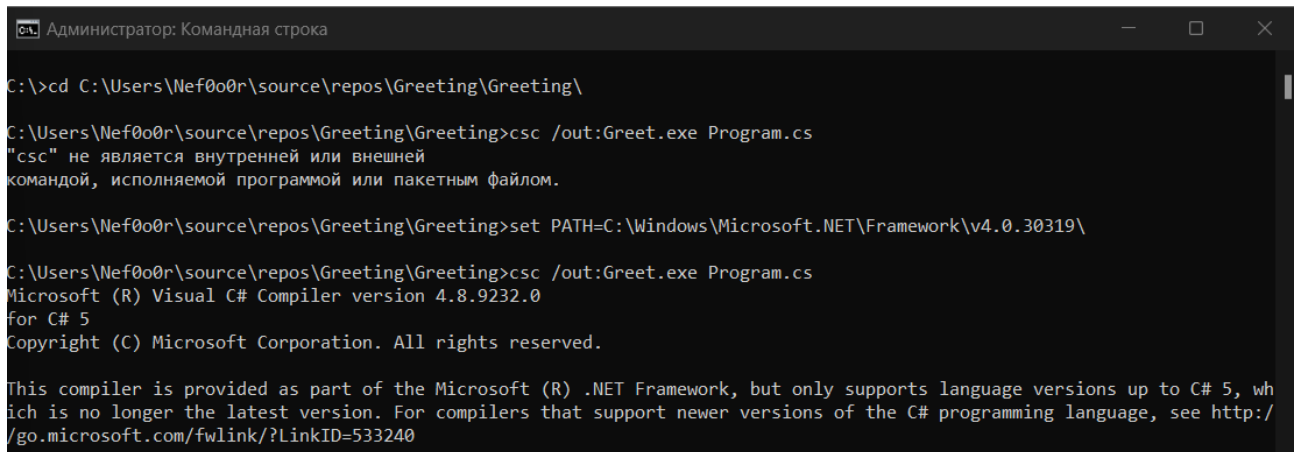


Рисунок 2 – Попытка запуска ранее написанной программы через встроенную КОНСОЛЬ

### 3. Отладка программы с использованием Visual Studio Debugger (рис. 3).

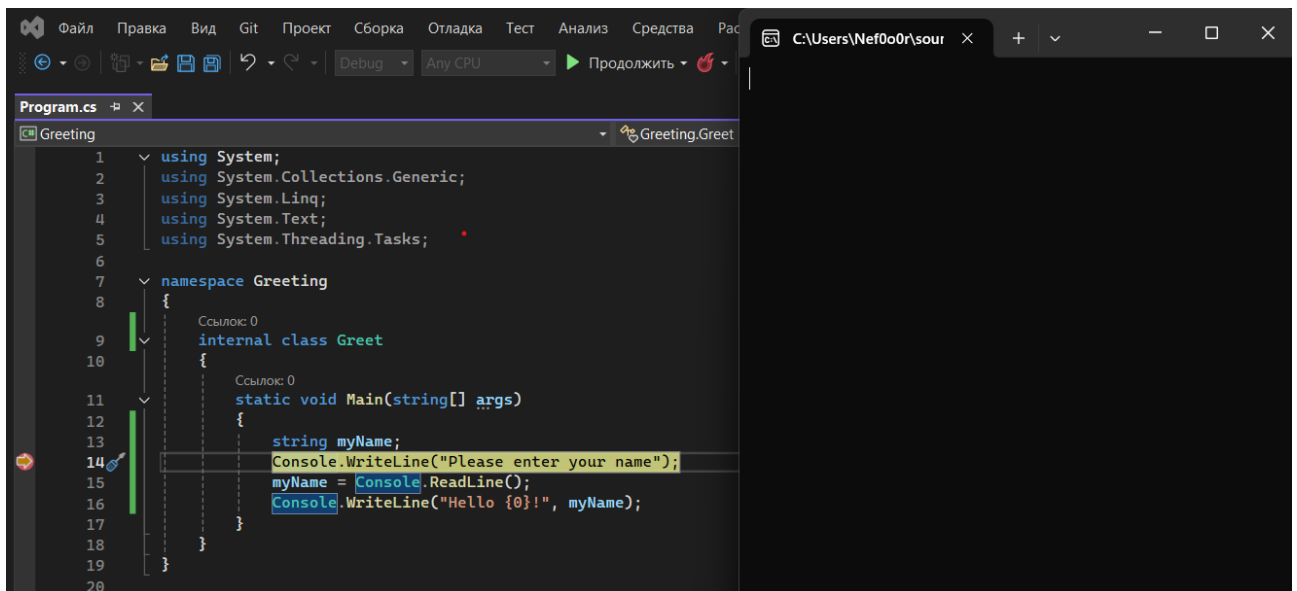


Рисунок 3 – Запуск кода через режим отладчика

После запуска кода через отладчик, был проверен режим пошаговой отладки (рис. 4).

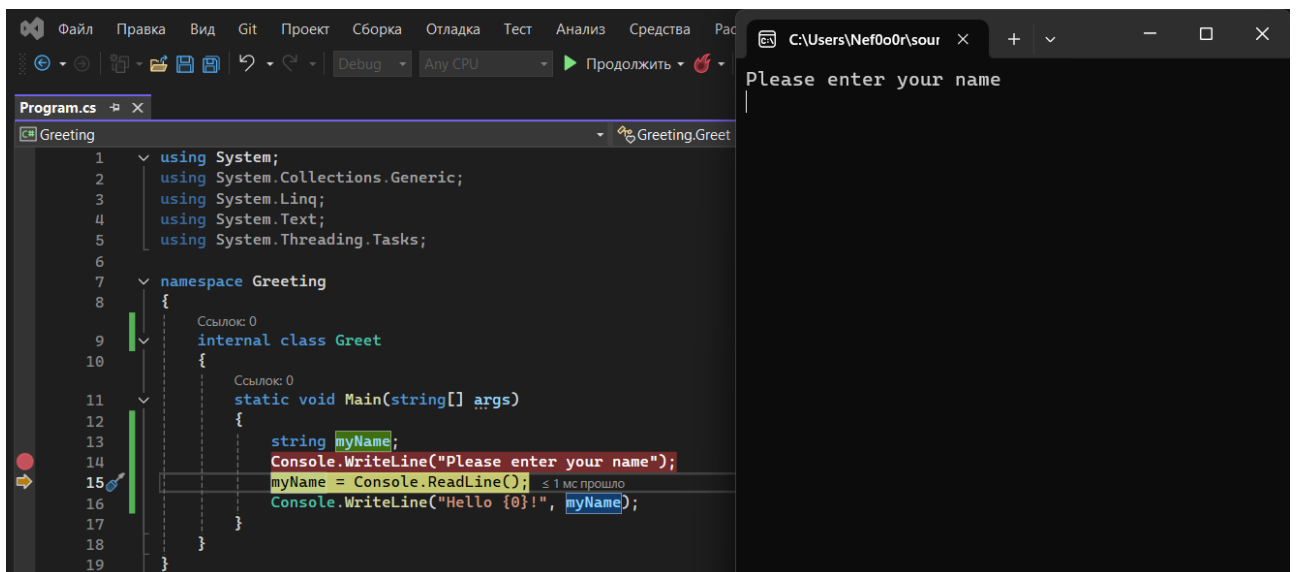


Рисунок 4 – Запуск кода по шагам через режим отладчика

#### 4. Добавление обработки исключений в программу на C#.

Был написан код, предназначенный для деления двух целых чисел, после чего он был протестирован в случае корректных (рис. 5) и некорректных значений (рис. 6).

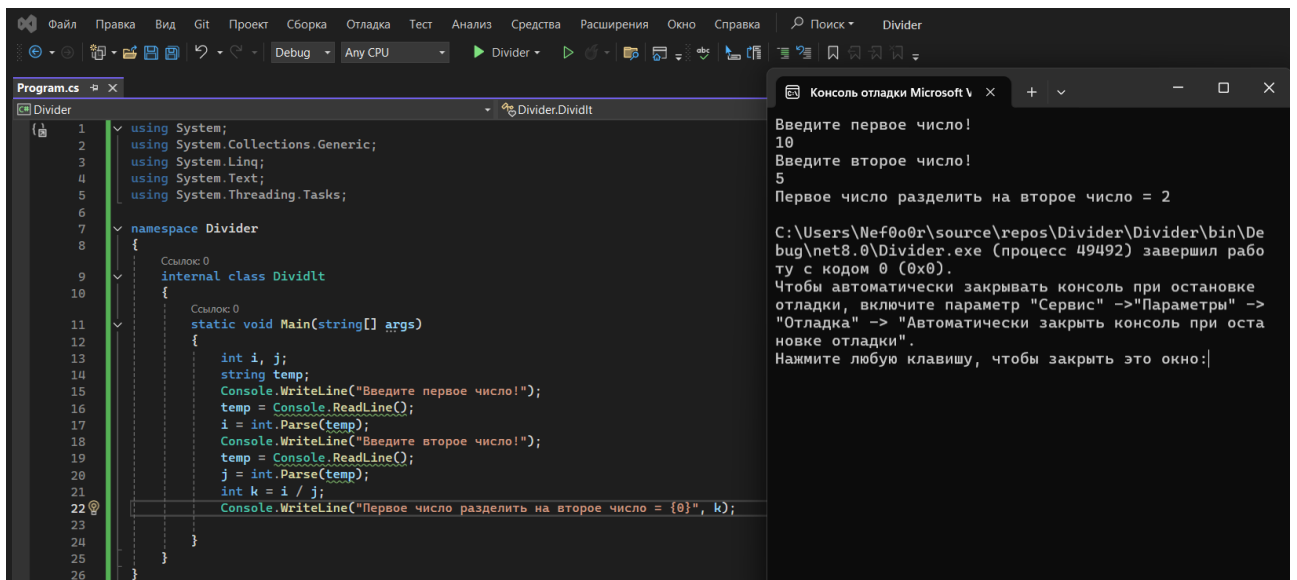


Рисунок 5 – Работа программы без блока для отслеживания исключений в случае отсутствия исключений

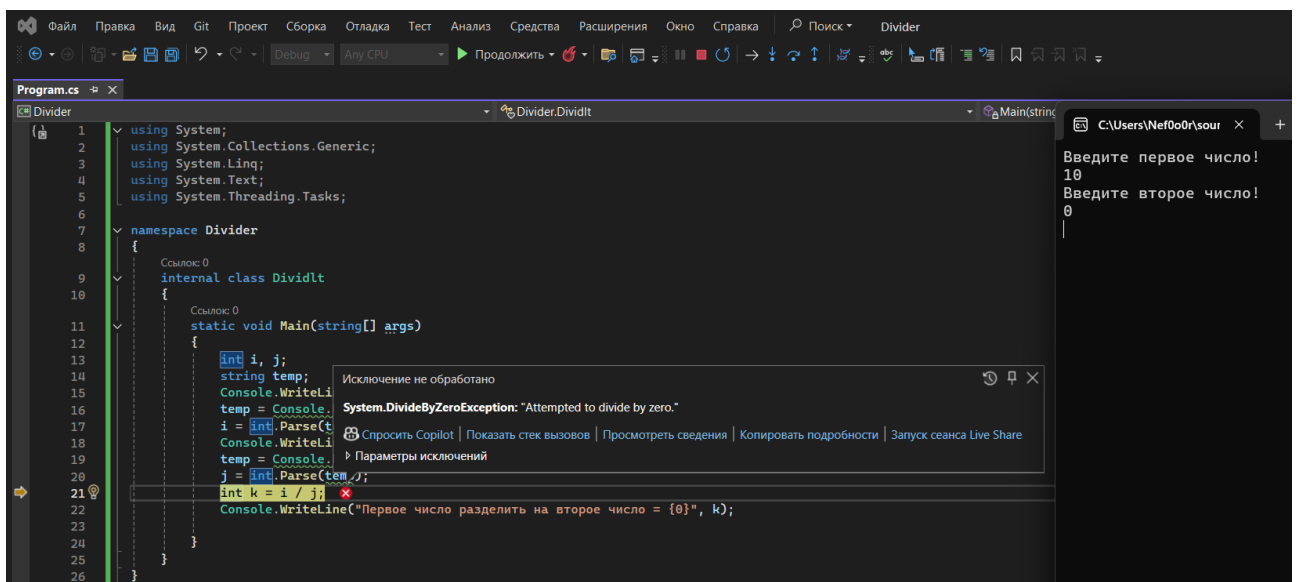


Рисунок 6 – Работа программы без блока для отслеживания исключений в случае возникновения исключений

После того, как было выяснено, что в случае ввода некорректных значений, код выдает исключение, необходимо было реализовать отслеживание подобных исключений. После чего данный код был повторно протестирован для корректных и некорректных значений.

В случае, корректных значений поведение кода не поменялось. В случае же, с некорректными значениями, код больше не завершался с ошибкой (рис. 7), а выводил соответствующее исключение в консоль (рис. 8).

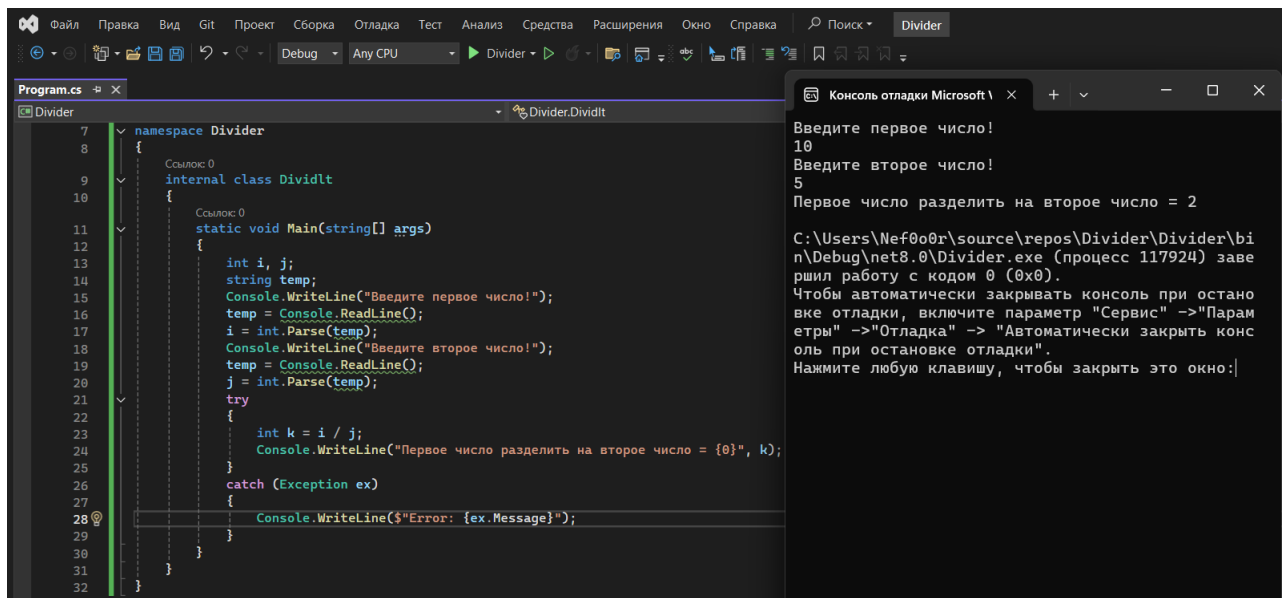


Рисунок 7 – Работа программы с отслеживанием исключений в случае отсутствия исключений

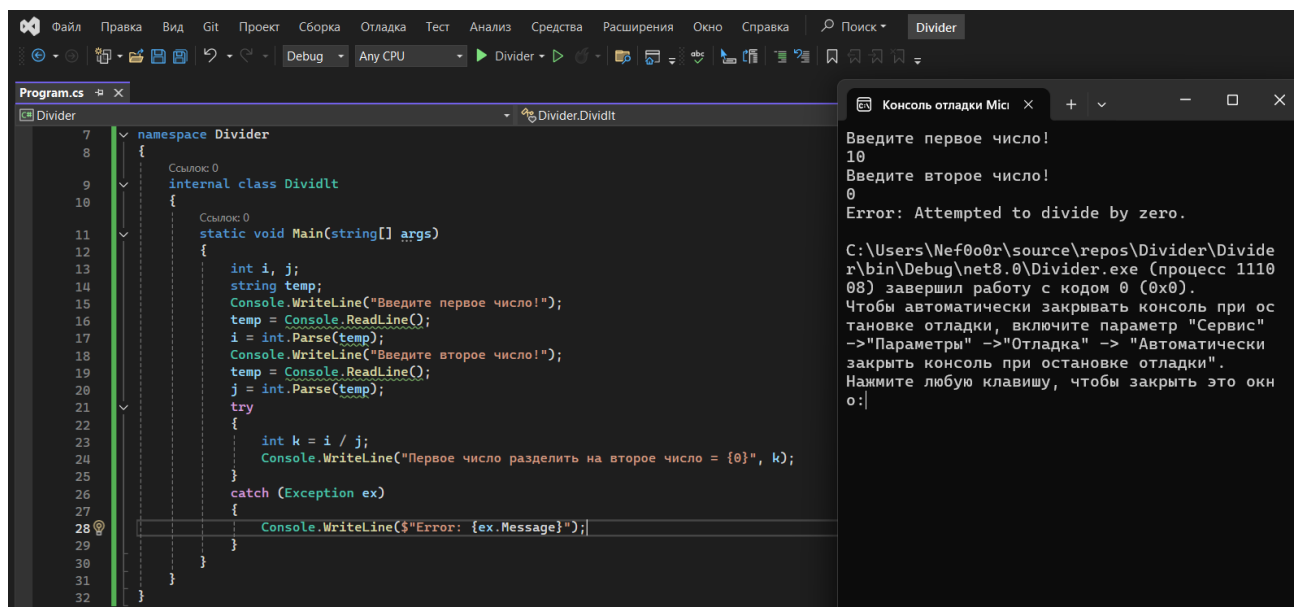


Рисунок 8 - Работа программы с отслеживанием исключений в случае наличия исключений

## Вывод

В ходе выполнения лабораторной работы были изучены базовые концепции разработки программ на языке C# с использованием интегрированной среды разработки Microsoft Visual Studio 2022. В рамках лабораторной работы были выполнены следующие упражнения:

1. Создание простой программы: Разработана консольная программа, которая запрашивает имя пользователя и выводит приветствие. Программа успешно создана, откомпилирована и запущена как из среды Visual Studio, так и с помощью командной строки.

2. Компиляция и запуск из командной строки: Изучены основы компиляции программ на C# через командную строку. Программа была не скомпилирована с помощью компилятора csc из-за отсутствия поддержки используемой версии C#.

3. Отладка программы: Освоены базовые навыки отладки программ с использованием встроенного отладчика Visual Studio. Программа была успешно запущена в режиме отладки, проведен пошаговый контроль выполнения команд.

4. Добавление обработки исключений: В программу добавлены конструкции для обработки исключений, такие как блоки try и catch. Программа успешно протестирована на корректные и некорректные входные данные, включая деление на ноль.

## ИСХОДНЫЙ КОД

### Упражнение 1:

```
using System;

namespace Greeting
{
    internal class Greet
    {
        static void Main(string[] args)
        {
            string myName;
            Console.WriteLine("Please enter your name");
            myName = Console.ReadLine();
            Console.WriteLine("Hello {0}!", myName);
        }
    }
}
```

### Упражнение 4:

```
using System;

namespace Divider
{
    internal class Dividlt
    {
        static void Main(string[] args)
        {
            int i, j;
            string temp;
            Console.WriteLine("Введите первое число!");
            temp = Console.ReadLine();
            i = int.Parse(temp);
            Console.WriteLine("Введите второе число!");
            temp = Console.ReadLine();
            j = int.Parse(temp);
            try
            {
                int k = i / j;
                Console.WriteLine("Первое число разделить на  
второе число = {0}", k);
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Error: {ex.Message}");
            }
        }
    }
}
```