# 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 С#

Студент гр. 3363	Минко Д.А., Овсейчик Н.И.
Преподаватель	Новакова Н.Е.

# Цель работы

Целью данной лабораторной работы является освоение основ программирования на языке С# 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 С# (рис. 1).

```
Файл Правка Вид Git Проект Сборка Обладка Тест Анализ Средства Расширения Окно Справка Р Поискт Greeting

Program.cs ** X

Greeting

vusing System;

using System. Collections. Generic;

using System. Linq;

using System. Text;

using System. Threading. Tasks;

6

7 namespace Greeting

8 {

Common O

static void Main(string[] args)

{

string myName;

Console. WriteLine("Hello {0}!", myName);

}

1 Name = Console. ReadLine();

Console. WriteLine("Hello {0}!", myName);

}

1 Name = Console. WriteLine("Hello {0}!", myName);

1 Name = Console. WriteLine("Hello {0}!", myName);
```

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

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

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

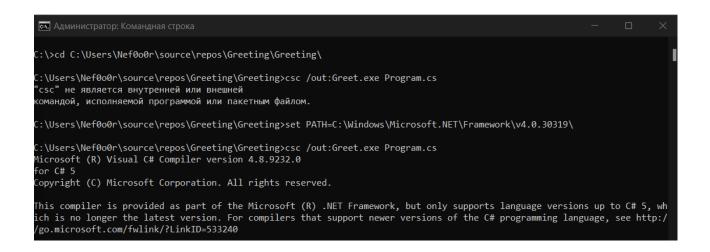


Рисунок 2 – Попытка запуска ранее написанной программы через встроенную консоль

3. Отладка программы с использованием Visual Studio Debuger (рис. 3).

Рисунок 3 – Запуск кода через режим отладчика

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

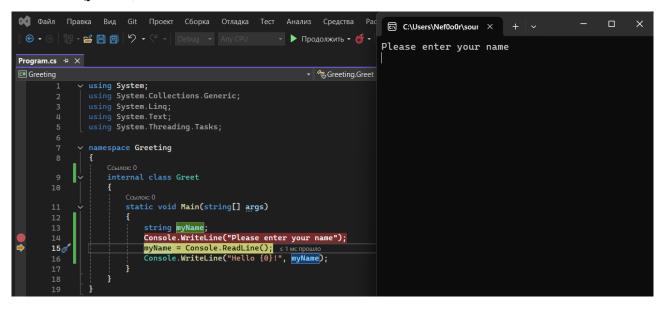


Рисунок 4 – Запуск кода по шагам через режим отладчика

4. Добавление обработки исключений в программу на С#.

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

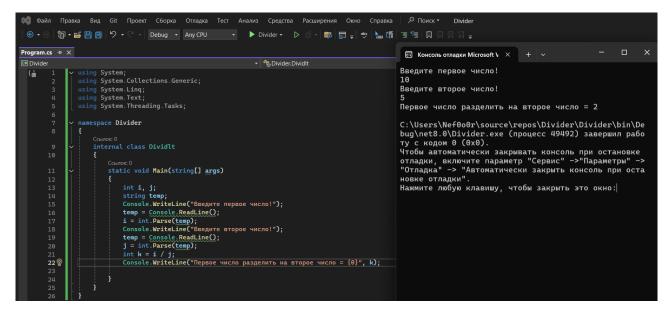


Рисунок 5 — Работа программы без блока для отслеживания исключений в случае отсутствия исключений

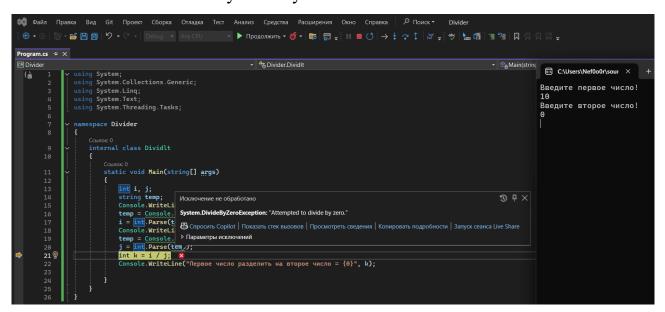


Рисунок 6 — Работа программы без блока для отслеживания исключений в случае возникновения исключений

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).

```
🙀 Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
             🏗 • 🛎 🖺 📳 🥠 • ୯ - Debug • Any CPU
                                                                                      - Divider - D グー 👼 🚮 🚽 💖 🔚 循 🖫 殭 😡 🕄
Program.cs → ×

    Консоль отладки Microsoft \ × + ∨

                                                                                                + %Divider.DividIt
                                                                                                                                                             Введите первое число!
                        namespace Divider
                                                                                                                                                             Введите второе число!
                              internal class Dividlt
                                                                                                                                                             Первое число разделить на второе число = 2
                                    static void Main(string[] args)
                                                                                                                                                            C:\Users\Nef0o0r\source\repos\Divider\Divider\bi
n\Debug\net8.0\Divider.exe (процесс 117924) заве
ршил работу с кодом 0 (0х0).
Чтобы автоматически закрывать консоль при остано
вке отладки, включите параметр "Сервис" ->"Парам
етры" ->"Отладка" -> "Автоматически закрыть конс
          11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28 §
                                         int i, j;
string temp;
Console.WriteLine("BBEDAUTE nepBoo MUCAO!");
temp = Console.ReadLine();
i = int.Parse(temp);
Console.WriteLine("BBEDAUTE BTOPOS MUCAO!");
temp = Console.ReadLine();
j = int.Parse(temp);
                                                                                                                                                             оль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:
                                               int k = i / j;
Console.WriteLine("Первое число разделить на второе число = \{0\}", k);
                                               Console.WriteLine($"Error: {ex.Message}");
```

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

```
📢 Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка 👂 Поиск 🕶
               🛅 🕶 🖺 📳 🥠 🕶 🖰 Debug 🕝 Any CPU
Program.cs 💠 🗙
                                                                                                              → %Divider.DividIt
☐ Divide
                                 espace Divider
                                                                                                                                                                                             Введите первое число!
                                                                                                                                                                                             Введите второе число!
                                  internal class Dividlt
                                                                                                                                                                                             Error: Attempted to divide by zero.
           11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28 29
29
30
                                                                                                                                                                                            C:\Users\Nef0o0r\source\repos\Divider\Divide r\bin\Debug\net8.0\Divider.exe (процесс 1110 08) завершил работу с кодом 0 (0x0). Чтобы автоматически закрывать консоль при ос тановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть коисоль при остановке отладки"
                                               int i, j;
string temp;
Console.WriteLine("Bведите первое число!");
temp = Console.ReadLine();
i = int.Parse(temp);
Console.WriteLine("Bведите второе число!");
temp = Console.ReadLine();
j = int.Parse(temp);
                                                                                                                                                                                             закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окн
                                                       int k = i / j;
Console.WriteLine("Первое число разделить на второе число = \{\theta\}", k);
                                                      Console.WriteLine($"Error: {ex.Message}");
```

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 С# 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 С# через командную строку. Программа была не скомпилирована с помощью компилятора сsc из-за отсутствия поддержки используемой версии С#.
- 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}");
                 }
             }
         }
     }
```