

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Отчет по лабораторной работе №1

по курсу «Объектно-ориентированное программирование»

Выполнил:

Влазнев Данила

К3223

Проверил:

Иванов С. Е.

Санкт-Петербург

2023 г.

## Содержание

Содержание .....	2
Упражнение 1 .....	3
Упражнение 2 .....	3
Упражнение 3 .....	4
Упражнение 4 .....	5
Упражнение 5 .....	6
Вывод .....	8

## Упражнение 1

В блокноте создается стандартная программа с точкой входа в виде функции Main. Считывается переменная и выводится текст. (См. Рисунок 1).

```
using System;

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

Рисунок 1 – Код для упражнения 1

Компилируем программу с помощью компилятора (См. Рисунок 2).  
Запускаем программу (См. Рисунок 3).

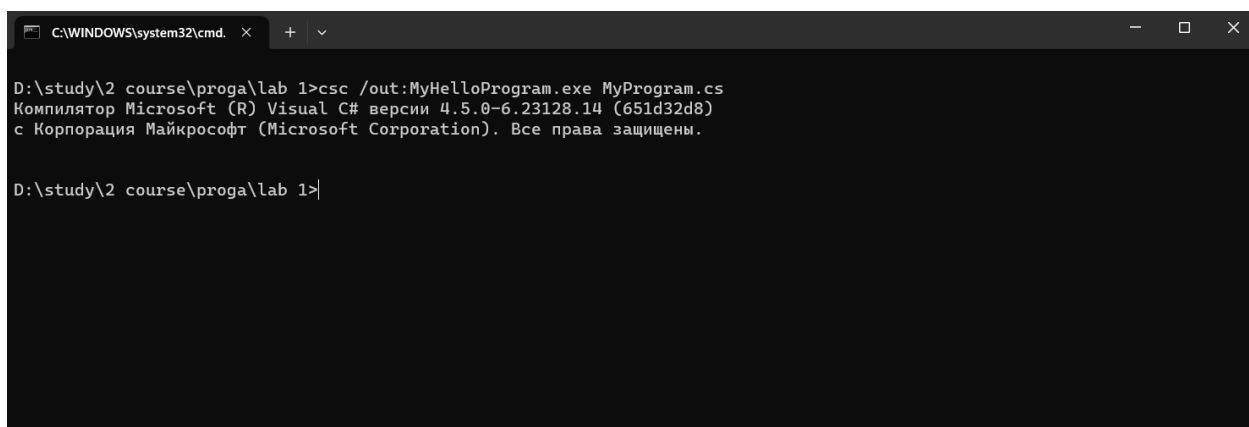


Рисунок 2 – Компиляция приложения



Рисунок 3 – Вывод программы

## Упражнение 2

В Visual Studio создается стандартная программа с точкой входа в виде функции Main. Считывается переменная и выводится текст. (См. Рисунок 4).  
Запускаем программу (См. Рисунок 5).

```

namespace Lab1
{
    Ссылка: 0
    internal class Greeter
    {
        Ссылка: 0
        static void Main(string[] args)
        {
            string myName;
            Console.WriteLine("Please enter your name");
            myName = Console.ReadLine();
            Console.WriteLine("Hello, {0}", myName);
        }
    }
}

```

Рисунок 4 – Код для упражнения 1

```

Please enter your name
Danila
Hello, Danila

D:\coding\works\с#\Lab1\Lab1\bin\Debug\net7.0\Lab1.exe (процесс 22464) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:|

```

Рисунок 5 – Вывод программы

### Упражнение 3

В данном упражнении происходит обучение использованию точек останова и отладки в целом. Поставим точку останова (См. Рисунок 6).

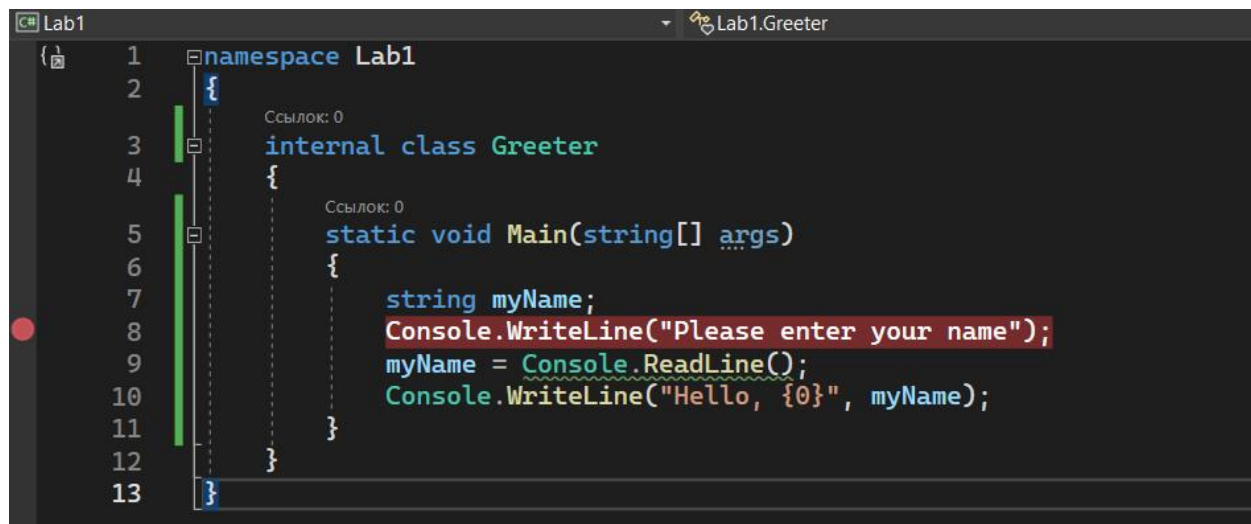


Рисунок 6 – Точка останова

Используя средства отладки, можно увидеть значение переменных до присвоения (См. Рисунок 7) и после (См. Рисунок 8).


Имя	Значение	Тип
 myName	null	string

Рисунок 7 – Значение переменной до присваивания



Имя	Значение	Тип
 System.Console.ReadLine возвраще...	"Danila"	string
 myName	"Danila"	string

Рисунок 8 – Значение переменной после присваивания

### Упражнение 4

Напишем программу для считывания 2 чисел и последующего целого деления первого на второе (См. Рисунок 9–10).

```
public static void Main(string[] args)
{
    try
    {
        Console.WriteLine("Please enter the first integer");
        string temp = Console.ReadLine();
        int i = Int32.Parse(temp);
        Console.WriteLine("Please enter the second integer");
        temp = Console.ReadLine();
        int j = Int32.Parse(temp);
        int k = i / j;
        Console.WriteLine("The result of dividing {0} by {1} is {2}", i, j, k);
    }
}
```

Рисунок 9 – Код программы

```
catch (FormatException e)
{
    Console.WriteLine("An format exception was thrown: {0}", e.Message);
}
catch (Exception e)
{
    Console.WriteLine("An exception was thrown: {0}",
        e.Message);
}
```

Рисунок 10 – Код программы

Примеры вывода программ при различном вводе (См. Рисунок 11–13).

```
Please enter the first integer
10
Please enter the second integer
5
The result of dividing 10 by 5 is 2
```

Рисунок 11 – Пример вывода

```
Please enter the first integer
10
Please enter the second integer
0
An exception was thrown: Attempted to divide by zero.
```

Рисунок 12 – Пример вывода

```
Please enter the first integer
ab
An format exception was thrown: The input string 'ab' was not in a correct format.
```

Рисунок 13 – Пример вывода

### Упражнение 5

В этом упражнении необходимо рассчитать сторону и площадь равностороннего треугольника по площади. Сначала считываем периметр, отлавливая ошибки, затем вычисляем необходимые поля. Выводим результаты с помощью форматирования (См. Рисунок 14–16).

```
static void Main(string[] args)
{
    Console.WriteLine("Введите периметр:");
    double perimeter;
    try
    {
        string temp = Console.ReadLine();
        perimeter = double.Parse(temp);
        if (perimeter <= 0)
        {
            throw new ArgumentException();
        }
    }
    catch (FormatException)
    {
        Console.WriteLine("Некорректный ввод");
        return;
    }
}
```

Рисунок 14 – Код программы

```
catch (ArgumentException)
{
    Console.WriteLine("Число должно быть больше 0");
    return;
}
double side = perimeter / 3;
double square = Math.Sqrt((perimeter / 2) * Math.Pow((perimeter / 2) - side, 3));
Console.WriteLine("{0, 7} {1}", "Сторона", "Площадь");
Console.WriteLine("{0, 7:0.##} {1, 7:0.##}", side, square);
```

Рисунок 15 – Код программы

```
Введите периметр:
18
Сторона  Площадь
        6    15,59
```

Рисунок 16 – Пример вывода программы

## **Вывод**

Я выполнил все задачи и научился создать первые программы на C#.