

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

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Объектно-ориентированное
программирование»
Тема: «Использование методов»

Студенты гр. 1302

_____ Романова О.В.

_____ Новиков Г.В.

Преподаватель:

_____ Васильев А.А.

Санкт-Петербург

2023

1. Цель работы

Изучение параметров в методах в языке C# с помощью программного продукта компании Microsoft VS 2022.

2. Анализ задачи

Необходимо:

- 1) Написать программу, которая сравнивает два целочисленных числа, введенных пользователем, и выводит большее;
- 2) Дополнить программу из первого упражнения, создать метод, который будет изменять значения параметров (параметры передаются по ссылке);
- 3) Дополнить программу, создать метод Factorial, который вычисляет факториал (у метода надо определить входной целочисленный параметр);
- 4) Дополнить программу рекурсивным вычислением факториала.

3. Ход выполнения работы

3.1 Упражнение 1

В ходе выполнения данного упражнения написана программа, которая сравнивает два целочисленных числа, введенных пользователем, и выводит то, которое больше.

3.1.1 Пошаговое описание алгоритма

Создать класс и метод, определить 3 целочисленных переменных x, y и greater.

Присвоить переменной greater большее значение переменных.

На экран пользователя вывести полученное число.

3.1.2 Используемые классы и методы

В программе, написанной в данном упражнении, используются следующие методы:

- `System.Console.WriteLine()` – служит для отображения на экране строк и значений переменных, переданных в метод в качестве параметров, с переходом на новую строку;

- `System.Console.ReadKey()` – ожидает следующего нажатия клавиши пользователем;

- `Main()` – служит для запуска программы.

- `Greater()` – возвращает значение наибольшего из двух чисел.

3.2 Упражнение 2

В ходе выполнения данного упражнения, написанная в предыдущем пункте программа, дополняется методом `swar`, меняющим местами значения.

3.2.1 Пошаговое описание алгоритма

Создать класс и метод, определить 3 целочисленных переменных `x`, `y` и `greater`.

Присвоить переменной `greater` большее значение переменных.

Поменять местами значения.

На экран пользователя вывести полученные числа.

3.2.2 Используемые классы и методы

В программе, написанной в данном упражнении, используются следующие методы:

- `System.Console.WriteLine()` – служит для отображения на экране строк и значений переменных, переданных в метод в качестве параметров, с переходом на новую строку;

- `System.Console.ReadKey()` – ожидает следующего нажатия клавиши пользователем [1];

- Main() – служит для запуска программы.

-Greater() – возвращает значение наибольшего из двух чисел.

-Swap() – меняет значения двух переменных местами.

3.3 Упражнение 3

В ходе выполнения данного упражнения, написанная в предыдущем пункте программа, дополняется методом factorial, вычисляющим факториал, числа введенного пользователем.

3.3.1 Пошаговое описание алгоритма

Создать класс и метод, определить 3 целочисленных переменных x, y и greater.

Присвоить переменной greater большее значение переменных.

Поменять местами значения.

Вычислить факториал.

На экран пользователя вывести полученные числа.

3.3.2 Используемые классы и методы

В программе, написанной в данном упражнении, используются следующие методы:

- System.Console.WriteLine() – служит для отображения на экране строк и значений переменных, переданных в метод в качестве параметров, с переходом на новую строку;

- System.Console.ReadKey() – ожидает следующего нажатия клавиши пользователем [1];

- Main() – служит для запуска программы.

-Greater() – возвращает значение наибольшего из двух чисел.

-Swap() – меняет значения двух переменных местами.

-Factorial() – вычисляет факториал числа.

3.4 Упражнение 4

В ходе выполнения данного упражнения, написанная в предыдущем пункте программа, дополняется методом RecursiveFactorial, вычисляющий факториал рекурсивно.

3.4.1 Пошаговое описание алгоритма

Создать класс и метод, определить 3 целочисленных переменных x, y и greater.

Присвоить переменной greater большее значение переменных.

Поменять местами значения.

Вычислить факториал.

На экран пользователя вывести полученные числа.

3.4.2 Используемые классы и методы

В программе, написанной в данном упражнении, используются следующие методы:

- System.Console.WriteLine() – служит для отображения на экране строк и значений переменных, переданных в метод в качестве параметров, с переходом на новую строку;

- System.Console.ReadKey() – ожидает следующего нажатия клавиши пользователем [1];

- Main() – служит для запуска программы.

-Greater() – возвращает значение наибольшего из двух чисел.

-Swap() – меняет значения двух переменных местами.

-Factorial() – вычисляет факториал числа.

-RecursiveFactorial() – рекурсивно вычисляет факториал.

3.4.3 Контрольный пример

На рис.3.1 представлены результаты выполнения программы.

```
Please enter number x:
5
Please enter number y:
7
Largest number: 7
x: 5, y: 7
After swap: x: 7, y: 5
Number for factorial:
5
Factorial 5 = 120
Factorial(rec) 5 = 120
```

Рис.3.1 Контрольный пример для программы

Как видно из рисунка, на экран выведены значения, большее из них и значения факториала, вычисленного двумя способами.

4. Листинг программы

```
using System;

namespace lab4
{
    class Utils
    {
        public static int Greater(int a, int b)
        {
            if (a > b)
                return a;
            else
                return b;
        }

        public static void Swap (ref int a, ref int b)
        {
            int temp = a;
            a = b;
            b = temp;
        }

        public static bool Factorial (int n, out int answer)
        {
            int i, f = 1;
            bool end = true;
            if (n >= 0)
            {
                try
                {
```

```

        for (i = 1; i <= n; i++)
        {
            f = f * i;
        }

        catch (Exception)
        {
            end = false;
        }

    else
        end = false;

    answer = f;

    return end;
}

public static int RecursiveFactorial(int n)
{
    int answer = 0;
    if (n == 0) return 1;
    try
    {
        if (n < 0) throw new ArgumentException("Factorial base must be >=
0");
        answer = RecursiveFactorial(n - 1) * n;
    }
    catch (Exception)
    {
        Console.WriteLine("Error");
        Environment.Exit(0);
    }
    return answer;
}

}

public class Test
{
    static void Main()
    {
        int x, y, z, greater, f;
        bool end;
        try
        {
            //first
            Console.WriteLine("Please enter number x: ");
            string num = Console.ReadLine();
            x = int.Parse(num);

            Console.WriteLine("Please enter number y: ");
            num = Console.ReadLine();
            y = int.Parse(num);

            greater = Utils.Greater(x, y);

            Console.WriteLine("Largest number: {0}", greater);

            //second
            Console.WriteLine("x: {0}, y: {1}", x, y);
            Utils.Swap(ref x, ref y);
            Console.WriteLine("After swap: x: {0}, y: {1}", x, y);

```

```

        //third
        Console.WriteLine("Number for factorial: ");
        num = Console.ReadLine();
        z = int.Parse(num);
        end = Utils.Factorial(z, out f);

        if (end)
            Console.WriteLine("Factorial {0} = {1}", z, f);
        else
            Console.WriteLine("Error");

        //fourth
        Console.WriteLine("Factorial(rec) {0} = {1}", z,
Utils.RecursiveFactorial(z));
    }
    catch (FormatException)
    {
        Console.WriteLine("Error");
    }
}
}
}

```

5. Полученные результаты

В ходе выполнения данной лабораторной работы нами были получены следующие результаты:

- в ходе работы программы были созданы методы и классы, с помощью которых вычислялось большее из значений, обмен значений местами и вычисление факториала.

6. Выводы

В ходе выполнения данной лабораторной работы:

- были изучены параметры методов в языке C#;
- была изучена передача параметров по ссылке в языке C#.
- были изучены рекурсивные функции в языке C#.

Список использованной литературы

1. MSDN — сеть разработчиков Microsoft. URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/method-parameters> (дата обращения: 05.03.2023)