Московский авиационный институт

(национальный исследовательский университет)

**Факультет информационных технологий и прикладной математики**

Кафедра вычислительной математики и программирования

**Лабораторная работа 7 по курсу ОOП:**

**основы программирования на языке С#**

ИСПОЛЬЗОВАНИЕ. КЛИЕНТ-СЕРВЕР. ОБЩНОСТЬ

Работу выполнил:

Березнев Никита Вадимович

Группа: М8О-203Б-21

Подпись: \_\_\_\_\_\_\_\_

Дата: 29 октября 2022 г.

**7.1 Клиент-сервер**

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CSharpLab7\_1

{

class Program

{

class F//клиент

{

public F() { Console.WriteLine("Constructor F"); }

~F() { }

public int j()

{

this.f = 5;

return this.f \* 2;

}

public int f { set; get; }

}

class K//сервер

{

public K() { Console.WriteLine("Constructor K"); }

~K() { }

public void mk(F f) { Console.WriteLine("class K client mk() = {0}", f.j()); }

}

static void Main(string[] args)

{

F f = new F();

K k = new K();

k.mk(f); // клиент(объект f) обращается к ресурсам сервера(класс K) через объект k с помощью метода mk.

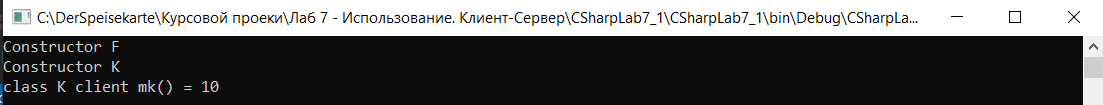
Console.ReadKey();

}

}

}

**Результат:**



**Вывод:**

С помощью использования клиент-сервер, можно контролировать или управлять одним объектом – клиентом, через другой объект – сервер. При это объект-клиент выступает в качестве параметра для объекта-сервера. Клиент и сервер могут находиться как на одном компьютере(ПК), так и на разных ПК в сети. Также может возникать такая ситуация, когда некоторый программный блок будет одновременно выполнять функции сервера по отношению к одному блоку и клиента по отношению к другому.

**7.2 Общность**

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CSharpLab7\_2

{

class Program

{

static class F

{

public static int J() { return 7; }

}

class K

{

public K() { }

~K() { }

public void mk() { Console.WriteLine("class F function J = {0}", F.J()); }

}

static void Main(string[] args)

{

K k = new K();

k.mk();

//нельзя сделать F f = new F();

F.J();

//Console.Readkey() — Console - класс. а ReadKey() - статическая функция

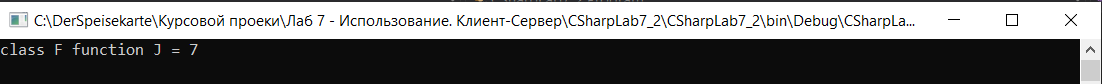
Console.ReadKey();

}

}

}

**Результат:**



**Вывод:**

Использование общности позволяет одному объекту взаимодействовать со всеми методами и членами второго, если первый объект по отношению ко второму объявляется другом. Также, чтобы открыть доступ функциям к данным объекта. они могут быть объявлены дружественными.