Тимофеева Наталья

М8О-202Б-19

ООП C#

Лабораторная работа № 7

**Описание работы**

ИСПОЛЬЗОВАНИЕ

1. Изучите категорию использование. Какие разновидности этой категории есть. Вершины графа, соединенные горизонтальной связью, определите на графе в виде категории использование, для этого поставьте соответствующие стрелки.

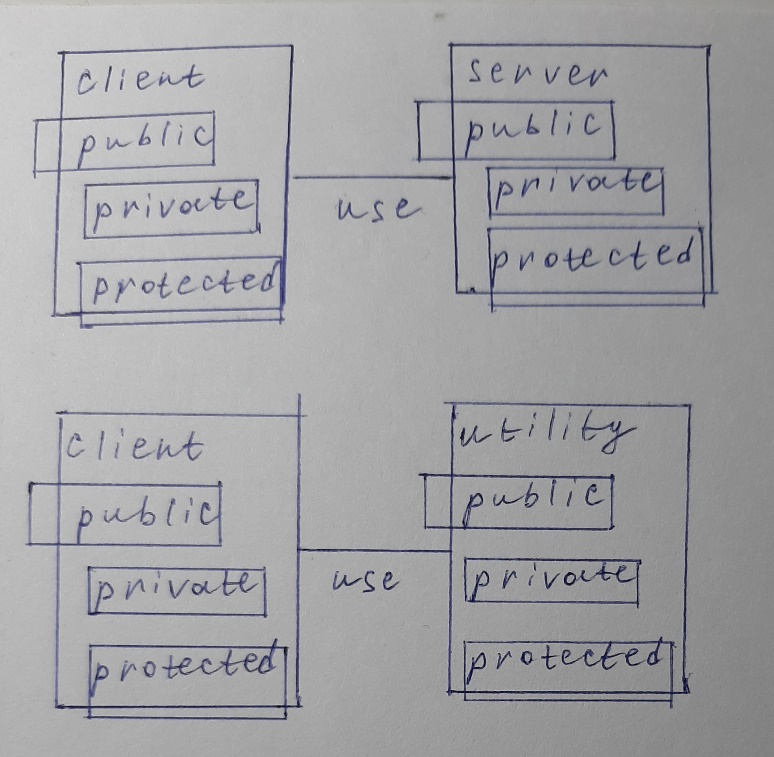
2. Реализуйте категорию использование клиент сервер, в отдельном проекте.

3. Реализуйте категорию использование общности. в отдельном проекте со скрытым конструктором.

4. Сравните полученные категории. В каком случае между объектами меньшее зацепление? В каком случае между объектами большая плотность?

5. Оформите работу. Проектам должны соответствовать графы. Напишите ответы на вопросы. Сохраните результаты лабораторной.

**Граф**



**Программа**

using System;

namespace Lab7 {

class Server {

public Server() { }

public int[] numbers = new int[10];

public void analysis(int a) {

if (a == 0) { ++numbers[0]; }

int copy = a;

while (a > 0) {

++numbers[a % 10];

a = a / 10;

}

for (int i = 0; i < 10; ++i) {

if (numbers[i] > 0) { Console.WriteLine("В числе {0} {1} цифр(а/ы) {2}", copy, numbers[i], i); }

}

}

public void func() { Console.WriteLine("Server method"); }

}

static class Utility {

static public void func() { Console.WriteLine("Utility method"); }

}

class Client {

public Client() { }

public void serv(Server s) {

Console.Write("Client requests a function from ");

s.func();

s.analysis(346937418);

}

public void utili() {

Console.Write("Client requests a function from ");

Utility.func();

}

}

class Program{

static void Main() {

Server s = new Server();

Client c = new Client();

c.serv(s);

c.utili();

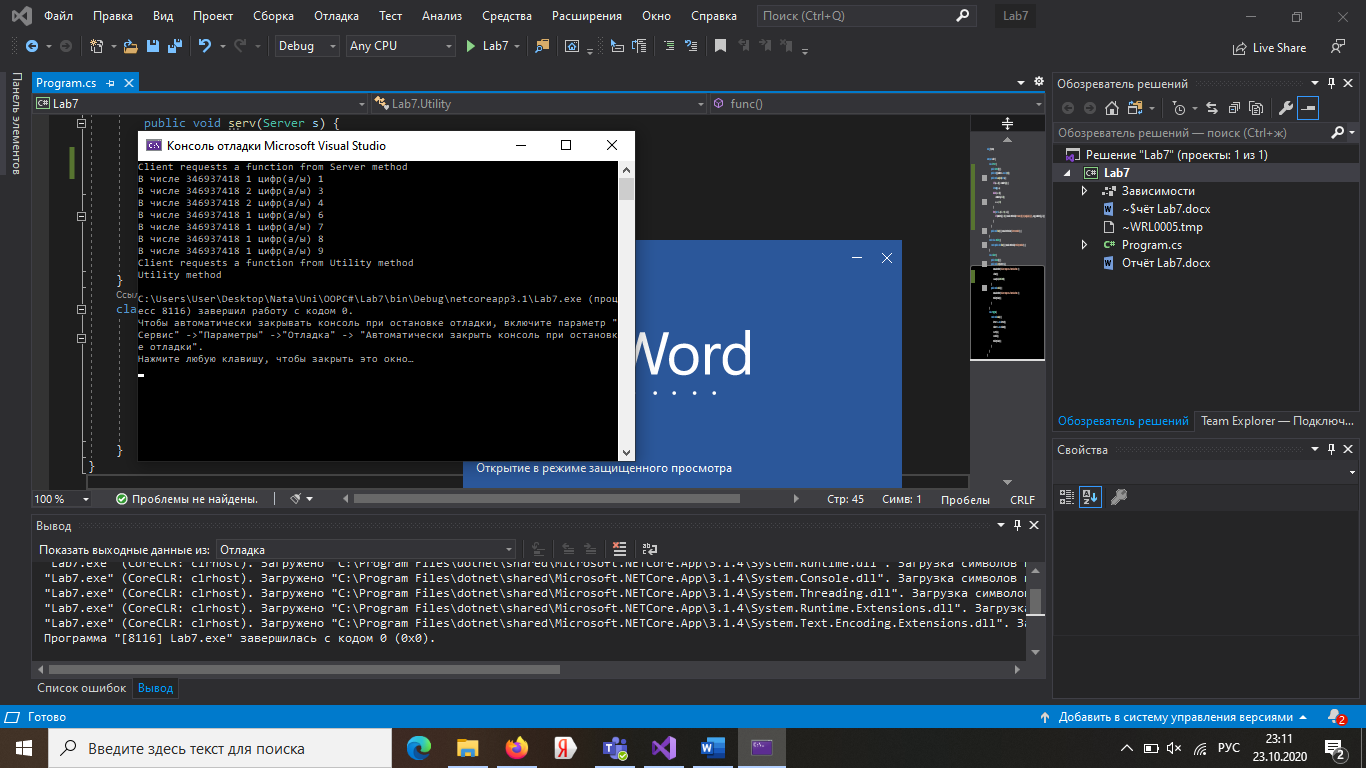
Utility.func();

}

}

}

**Вывод с консоли**



**Вывод**

С помощью использования клиент-сервер, можно управлять одним классом – клиентом через другой класс – сервер. Клиент вызывает функции сервера. Также в сервере могут храниться внутренние данные.

Использование общности позволяет одному классу взаимодействовать со всеми функциями второго. Владельцем данных является тот объект, который делает вызов. Вызываемый класс делаем статическим.