Тимофеева Наталья

М8О-202Б-19

ООП C#

Лабораторная работа № 5

**Описание работы**

НАСЛЕДОВАНИЕ: КОМБИНИРОВАНИЕ ЧЕРЕЗ ОБЩИХ ПРЕДКОВ

1. На графе выделите вершину, являющуюся общим предком так, чтобы в выделенном графе было множественное наследование. При необходимости дополните граф требуемой связью. Реализуйте категорию наследование комбинирование через общих предков.

2. Рассмотрите комбинацию конструкторов и применение модификатора virtual при множественном наследовании.

Сравните варианты наследования с модификатором virtual и без него. В каком случае выполняется и не выполняется принцип подстановки?

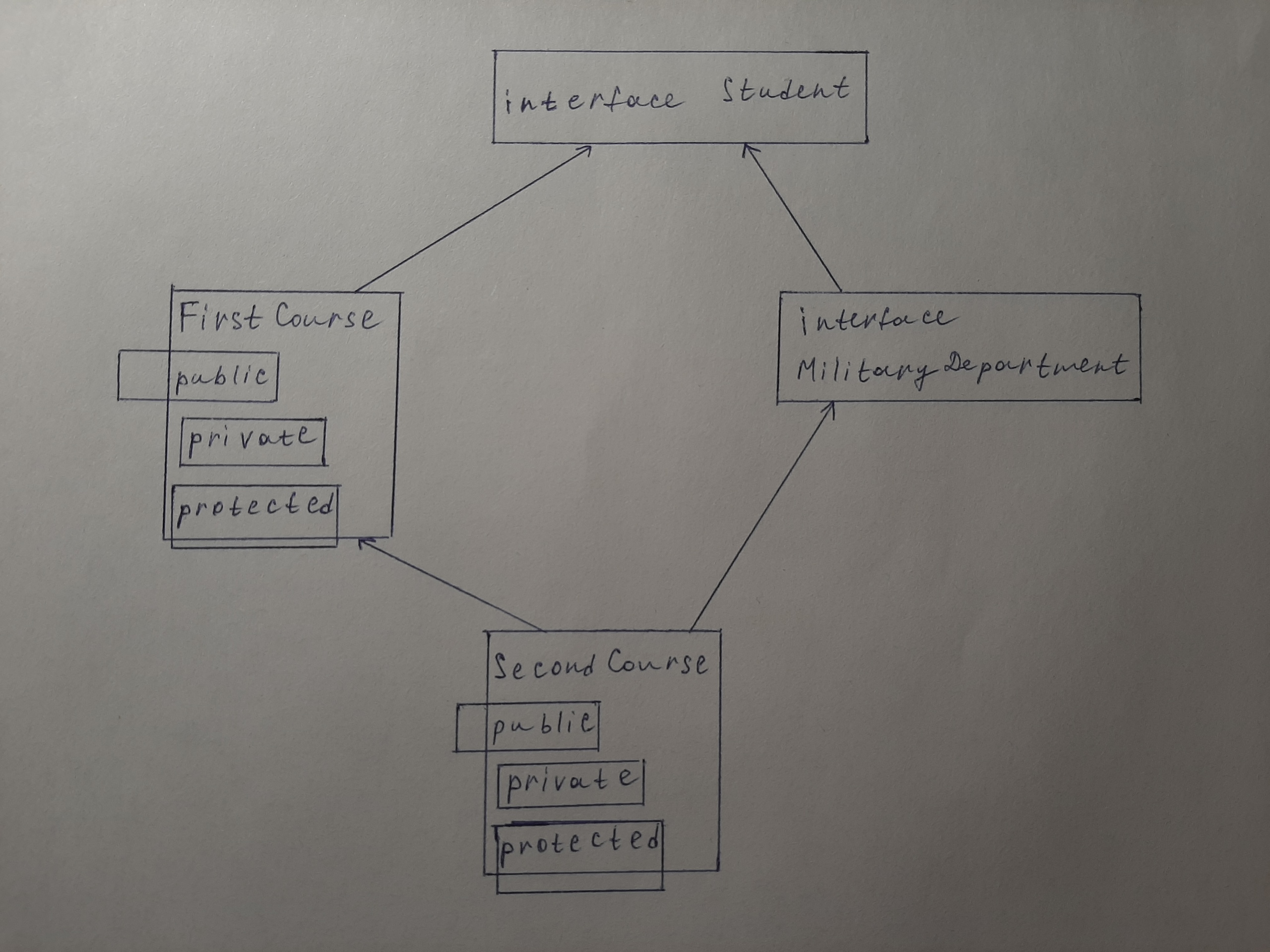
Какая зависимость между конструкторами создаваемых объектов, на что влияет их прядок, при изменении его в подклассах?

Напишите достоинства и недостатки данной категории?

Сравните известные виды иерархии. В чем их принципиальное отличие?

3. Оформите работу. Проекту должен соответствовать граф, реализованный в виде категории наследование комбинирование через общих предков. Напишите ответы на вопросы к пункту 2. Сохраните результаты лабораторной.

**Граф**



**Программа**

using System;

namespace Lab5 {

interface Student {

void lost();

}

interface MilitaryDepartment : Student {

void enteredToMD();

}

class FirstCourse : Student {

public int maths, informatics, history;

public FirstCourse() {

maths = 5;

informatics = 5;

history = 5;

}

public FirstCourse(int a, int b, int c) {

maths = a;

informatics = b;

history = c;

}

public void lost() { Console.WriteLine("не знает, где находятся аудитории"); }

public void money1() {

if ((maths >= 4) && (informatics >= 4) && (history >= 4)) {

Console.WriteLine("получает стипендию на первом курсе");

}

else { Console.WriteLine("не получает стипендию на первом курсе"); }

}

}

class SecondCourse : FirstCourse, MilitaryDepartment {

public int OOP, OS, DA;

public SecondCourse() {

OOP = 5;

OS = 5;

DA = 5;

}

public SecondCourse(int a1, int b1, int c1, int a2, int b2, int c2) : base(a1, b1, c1) {

OOP = a2;

OS = b2;

DA = c2;

}

public void lost() { Console.WriteLine("знает, где находятся аудитории"); }

public void enteredToMD() {

if ((maths == 5) && (informatics == 5) && (history >= 4)) {

Console.WriteLine("поступил на военную кафедру");

}

else { Console.WriteLine("не поступил на военную кафедру"); }

}

public void money2() {

if ((OOP >= 4) && (OS >= 4) && (DA >= 4)) {

Console.WriteLine("получает стипендию на втором курсе");

}

else { Console.WriteLine("не получает стипендию на втором курсе"); }

}

}

class Program {

static void Main() {

Student Tanya = null;

Tanya = new FirstCourse();

Console.WriteLine("Таня");

Tanya.lost();

Student Masha = null;

Masha = new SecondCourse();

Console.WriteLine("Маша");

Masha.lost();

FirstCourse Vasya = new FirstCourse(3, 4, 5);

Console.WriteLine("Вася");

Vasya.lost();

Vasya.money1();

SecondCourse Sasha = new SecondCourse(5, 5, 4, 5, 4, 3);

Console.WriteLine("Саша");

Sasha.lost();

Sasha.enteredToMD();

Sasha.money1();

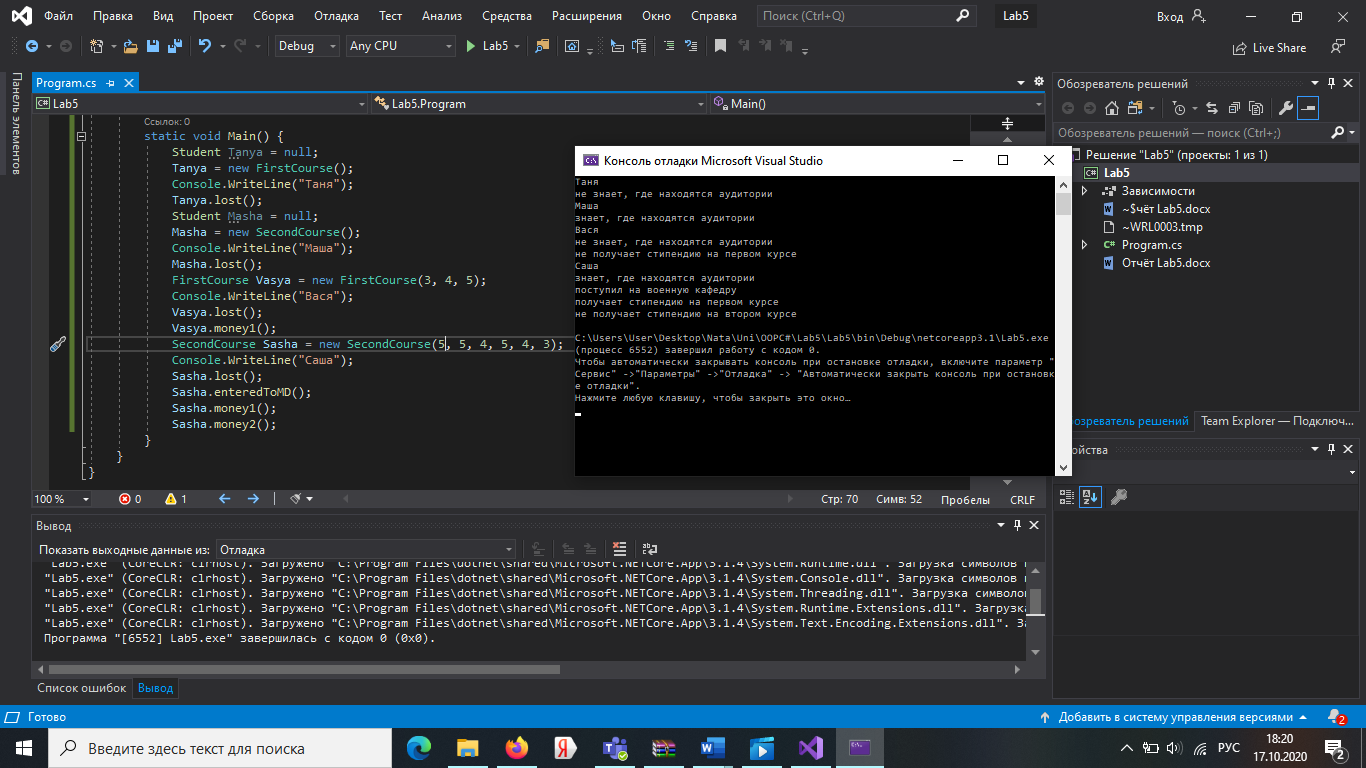
Sasha.money2();

}

}

}

**Вывод с консоли**



**Вывод**

Я научилась реализовывать наследование через общих предков. Наследование через общих предков комбинирует в себе класс и интерфейс с общим интерфейсом.