Московский Авиационный Институт

(Национальный Исследовательский Университет)

**Факультет информационных технологий и прикладной математики**

Кафедра вычислительной математики и программирования

Лабораторная работа 3 по курсу ООП:

Основы программирования на языке C#

3.ПРИНЦИП ПОДСТАНОВКИ

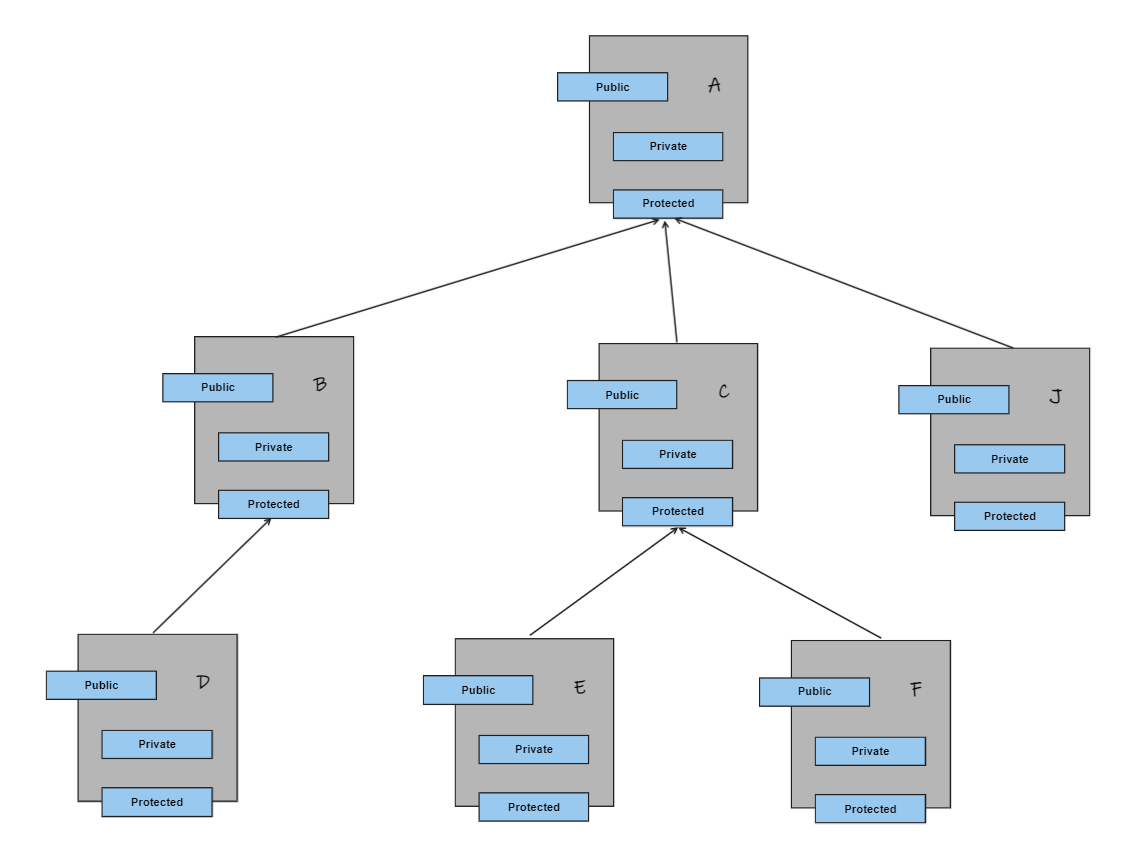
Работу выполнил:

М8О-205Б-21 Жилов А.А. \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

Руководитель: \_\_\_\_\_\_\_\_\_\_\_/Кузнецова С.В.

Дата: \_\_\_Октября 2022

**Принцип подстановки**

****

**Текст программы**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab\_3

{

public class A

{

public A()

{

Console.WriteLine("Constructor A");

this.varA = 0;

}

~A() { }

public virtual int Fm()

{

Console.WriteLine("Func class A\t a=0");

return this.varA + this.varA;

}

protected int varA { get; set; }

}

public class J:A

{

public J()

{

Console.WriteLine("Constructor J");

this.varJ = 1;

}

~J() { }

public override int Fm()

{

Console.WriteLine("Func of class J\t j=1");

return this.varA + this.varJ;

}

protected int varJ { get; set; }

}

public class B : A

{

public B()

{

Console.WriteLine("Constructor B");

this.varB = 10;

}

~B() { }

public override int Fm()

{

Console.WriteLine("Func of class B\t b=10");

return this.varB + this.varA;

}

protected int varB { get; set; }

}

public class C:A

{

public C()

{

Console.WriteLine("Constructor C");

this.varC = 7;

}

~C() { }

public override int Fm()

{

Console.WriteLine("Func of class C\t c=7");

return this.varC + this.varA;

}

protected int varC { get; set; }

}

public class D : B

{

public D()

{

Console.WriteLine("Constructor D");

this.varD = 75;

}

~D() { }

public override int Fm()

{

Console.WriteLine("Func of class K\t d=75");

return this.varD + this.varB;

}

protected int varD { get; set; }

}

public class E : C

{

public E()

{

Console.WriteLine("Constructor E");

this.varE = 45;

}

~E() { }

public override int Fm()

{

Console.WriteLine("Func of class E\t e=45");

return (this.varC \* this.varA \* 2 + this.varE);

}

protected int varE { get; set; }

}

class F : C

{

public F()

{

Console.WriteLine("Constructor F");

this.varF = 9;

}

~F() { }

public override int Fm()

{

Console.WriteLine("Func of class F\t f=9");

return (this.varC \* this.varA \* 4 + this.varF);

}

protected int varF { get; set; }

}

class Program

{

static void Main(string[] args)

{

A a = new A();

Console.WriteLine($"a.Fm() = {a.Fm()}");

a = new B();

Console.WriteLine($"a.Fm() = {a.Fm()}");

a = new C();

Console.WriteLine($"a.Fm() = {a.Fm()}");

a = new J();

Console.WriteLine($"a.Fm() = {a.Fm()}");

a = new D();

Console.WriteLine($"a.Fm() = {a.Fm()}");

a = new E();

Console.WriteLine($"a.Fm() = {a.Fm()}");

a = new F();

Console.WriteLine($"a.Fm() = {a.Fm()}");

}

}

}

**Результат работы  
Вывод**

В этой программе используется метод подстановки и метод замещения. **Принцип подстановки:** вместо объекта суперкласса можно подставить объект подкласса. **Принцип замещения:** функцию суперкласса можно заменить функцией подкласса. Ключевое слово virtual используется для изменения объявлений методов, свойств, индексаторов и событий, и разрешения их переопределения в производном классе. Например, этот метод может быть переопределен любым наследующим его классом: модификатор override требуется для расширения или изменения абстрактной или виртуальной реализации унаследованного метода, свойства, индексатора или события.