Московский авиационный институт

(национальный исследовательский университет)

**Факультет информационных технологий и прикладной математики**

Кафедра вычислительной математики и программирования

**Лабораторная работа 3 по курсу ОOП:**

**основы программирования на языке С#**

ПРИНЦИП ПОДСТАНОВКИ

Работу выполнил:

Березнев Никита Вадимович

Группа: М8О-203Б-21

Подпись: \_\_\_\_\_\_\_\_

Дата: 29 октября 2022 г.

**3. Принцип подстановки**



**Код программы:**

using System;

namespace CSharpLab3

{

class Program

{

class A

{

public A()

{

this.varA = 5;

}

~A() { }

public virtual int F()

{

Console.WriteLine("Func class A");

return this.varA + 30;

}

protected int varA { get; set; }

}

class C : A

{

public C()

{

this.varA = 15;

this.varC = 2;

}

~C() { }

public override int F()

{

Console.WriteLine("Func of class C");

return this.varA + 30;

}

public virtual int F1()

{

Console.WriteLine("Func of class C");

return (this.varC \* this.varA);

}

protected int varC { get; set; }

}

class E : C

{

public E() { }

~E() { }

public override int F1()

{

Console.WriteLine("Func of class E");

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

}

}

static void Main(string[] args)

{

Console.WriteLine("Подстановка и расширение функции F класса А:");

A a1 = new A();

Console.WriteLine("a1.F() = {0}", a1.F());

a1 = new C();

Console.WriteLine("a1.F() = {0}", a1.F());

Console.WriteLine();

Console.WriteLine("проверка, является ли a1 объектом класса С:");

if (a1 is C)

{

Console.WriteLine("a1 объект класса C");

}

else Console.WriteLine("а1 не является объектом класса C");

Console.WriteLine();

{

C c = new C();

int k = c.F();

Console.WriteLine(k);

}

Console.WriteLine("Расширение функции F1 класса C:");

C c1 = new C();

Console.WriteLine("c1.F1() = {0}", c1.F1());

c1 = new E();

Console.WriteLine("c1.F1() = {0}", c1.F1());

Console.WriteLine();

Console.WriteLine("c1.F() = {0}", c1.F());

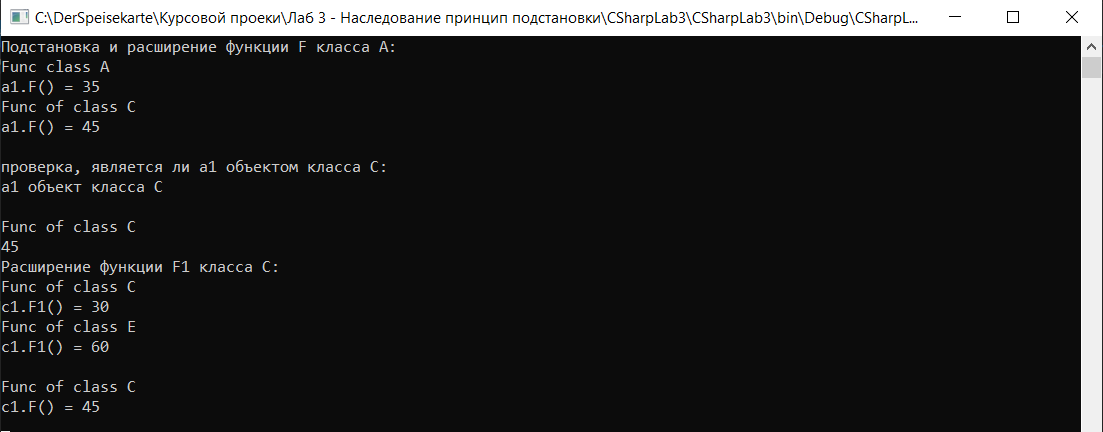
Console.ReadKey();

}

}

}

**Результат:**



**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.