Московский Авиационный Институт

(Национальный Исследовательский Университет)

**Факультет информационных технологий и прикладной математики**

Кафедра вычислительной математики и программирования

Лабораторная работа 5 по курсу ООП:

Основы программирования на языке C#

5. КОМБИНИРОВАНИЕ ЧЕРЕЗ ОБЩИХ ПРЕДКОВ.

Работу выполнил:

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

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

Дата: \_\_\_Октября 2022

**КОМБИНИРОВАНИЕ ЧЕРЕЗ ОБЩИХ ПРЕДКОВ.**

**Текст программы**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Reflection.Metadata.Ecma335;

using System.Text;

using System.Threading.Tasks;

namespace lab\_5

{

public interface A

{

void mIA();

int fIA();

}

public class B:A

{

private int b = 9;

public B()

{

Console.WriteLine("Construcrtor of B(no argument)");

this.b = 1;

}

public B(int b)

{

Console.WriteLine("Construcrtor of B(argument)");

this.b = b;

}

public int fAS() { return this.b; }

public void mIA() { }

public int fIA() { return 21; }

}

public interface C:A

{

int fIC();

void mIC();

}

public class E:B,C

{

public E()

{

this.a = this.fAS();

}

public E(int a,int b):base(b)

{

Console.WriteLine("Constructor of E");

this.a = a;

}

public void mIC() { this.a = this.fIC() + this.fIA(); }

public int fIC() { return 12; }

private int a=0;

}

class Programm

{

static void Main(string[] args)

{

A a = null;

a = new B();

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

C c = null;

c = new E();

Console.WriteLine($" C c.fIA() ={ c.fIA() + c.fIC()}");

a = new E();

a = new E(8, 5);

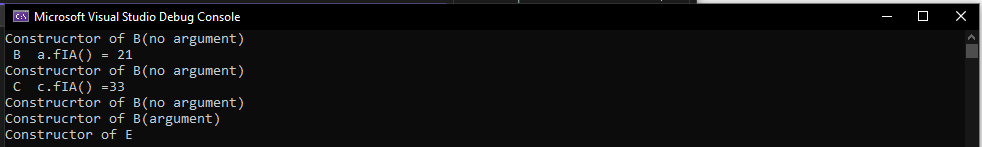
Console.ReadKey();

}

}

}

**Результат работы**

****

**Вывод**

Позволяет реализовать множественное наследование через общий супер-класс; комбинирует в себе два класса с общим супер-классом, который, в свою очередь, сам приутствует только в единственном экземпляре благодаря virtual наследованию, позволяющее избежать дублирование этого супер-класса.