Московский авиационный институт

(национальный исследовательский университет)

**Факультет информационных технологий и прикладной математики**

Кафедра вычислительной математики и программирования

**Лабораторная работа 1 по курсу ОOП:**

**основы программирования на языке С#**

Агрегация по ссылке

Работу выполнил:

Березнев Никита Вадимович

Группа: М8О-203Б-21

Подпись: \_\_\_\_\_\_\_\_

Дата: 29 октября 2022 г.



**Код программы:**

using System;

namespace CSharpLab1

{

class Program

{

class A

{

private C c = null;

private J j = null;

public A(C c, J j)

{

this.c = c;

this.j = j;

c.cq = 21;

}

~A() { }

public void mA()

{

Console.WriteLine("method of A");

}

public J jA

{

set { Console.WriteLine("set j"); j = value; }

get { Console.Write("get j ->"); return j; }

}

public C cA

{

set { Console.WriteLine("set c"); c = value; }

get { Console.Write("get c ->"); return c; }

}

}

class C

{

D d = null;

E e = null;

F f = null;

public C(D d, E e, F f)

{

this.d = d;

this.e = e;

this.f = f;

}

~C() { }

public void mC()

{

Console.WriteLine("method of C");

}

public D dA

{

set { Console.WriteLine("set d"); d = value; }

get { Console.Write("get d ->"); return d; }

}

public E eA

{

set { Console.WriteLine("set e"); e = value; }

get { Console.Write("get e ->"); return e; }

}

public F fA

{

set { Console.WriteLine("set f"); f = value; }

get { Console.Write("get f ->"); return f; }

}

public int cq { get; set; }

}

class J

{

K k = null;

public J(K k)

{

this.k = k;

}

~J() { }

public void mJ()

{

Console.WriteLine("method of J");

}

public K kA

{

set { Console.WriteLine("set k"); k = value; }

get { Console.Write("get k ->"); return k; }

}

}

class D

{

public D() { }

~D() { }

public void mD()

{

Console.WriteLine("Method of D");

}

}

class E

{

public E() { }

~E() { }

public void mE()

{

Console.WriteLine("Method of E");

}

}

class F

{

public F() { }

~F() { }

public void mF()

{

Console.WriteLine("Method of F");

}

}

class K

{

public K() { }

~K() { }

public void mK()

{

Console.WriteLine("Method of K");

}

}

static void Main(string[] args)

{

D d = new D();

E e = new E();

F f = new F();

K k = new K();

C c = new C(d, e, f);

J j = new J(k);

A a = new A(c, j);

Console.WriteLine("печать атрибута доступа:");

Console.WriteLine(c.cq);

Console.WriteLine("передача по ссылке:");

a.mA();

a.cA.mC();

a.jA.mJ();

a.cA.dA.mD();

a.cA.eA.mE();

a.cA.fA.mF();

a.jA.kA.mK();

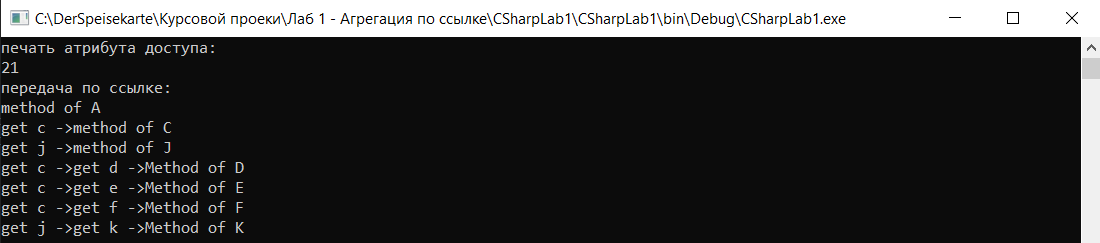
Console.ReadKey();

}

}

}

**Результат:**



**Вывод:**

Объекты всех классов существуют независимо друг от друга. Связывание объектов происходит  с помощью конструктора. Например, *b, j,  —* параметры для конструктора класса A; *d* — для конструктора класса B; *e, f,k —* для конструктора класса C. Объекты могут быть уничтожены по отдельности. Это нарушит целостность структуры. Если удалить объект *a,* объекты *b, j* и т.д. будут продолжать существовать и дальше. Агрегация по ссылке очень выгодна, так как она расходует очень мало памяти.