

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 Титеев Р.М.

(подпись)

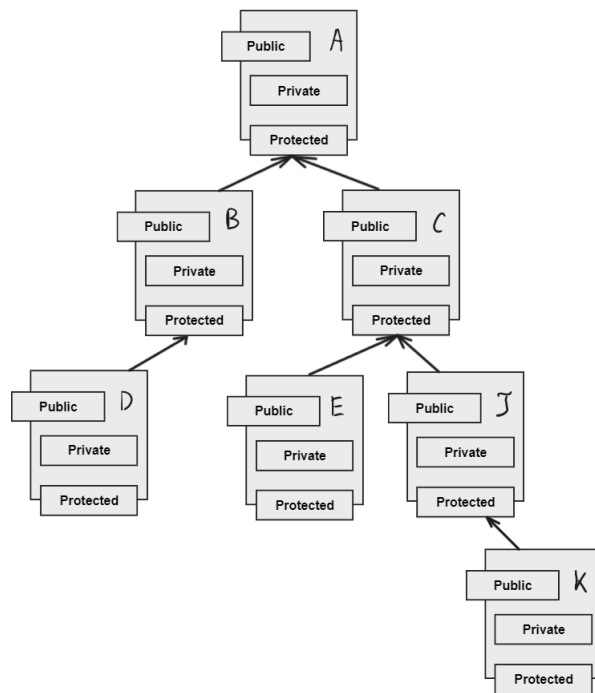
(вариант)

Руководитель: _____/Кузнецова С.В.

(подпись)

Дата: ____ октября 2022

Принцип подстановки



Текст программы

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace lab_3
8 {
9     public class A {
10         public A(){
11             Console.WriteLine("constructor A ");
12             this.count = 0;
13         }
14         protected int count { set; get; }
15         public virtual int fm() {
16             return this.count + 1;
17         }
18     }
19     public class B : A {
20         public B(){
21             Console.WriteLine("constructor B ");
22             this.count = 1;
23         }
24         public override int fm() {
```

```

25         return this.count * 2;
26     }
27 }
28 public class D : B {
29     public D(){
30         Console.WriteLine("constructor D ");
31         this.count = 2;
32     }
33     public override int fm() {
34         return this.count + 2;
35     }
36 }
37
38 public class C : A {
39     public C(){
40         Console.WriteLine("constructor C ");
41         this.count = 1;
42         this.mult = 10;
43     }
44     protected int mult { set; get; }
45     public override int fm() {
46         return this.count * this.mult;
47     }
48 }
49 public class E : C {
50     public E(){
51         Console.WriteLine("constructor E ");
52         this.count = 1;
53         this.mult = 2;
54     }
55     public override int fm() {
56         return this.count + 3*this.mult;
57     }
58 }
59 public class J : C {
60     public J(){
61         Console.WriteLine("constructor J ");
62         this.count = 1;
63         this.mult = 5;
64     }
65     public override int fm() {
66         return this.count * 3 + this.mult;
67     }
68 }
69 public class K : J {
70     public K(){
71         Console.WriteLine("constructor K ");
72         this.count = 1;

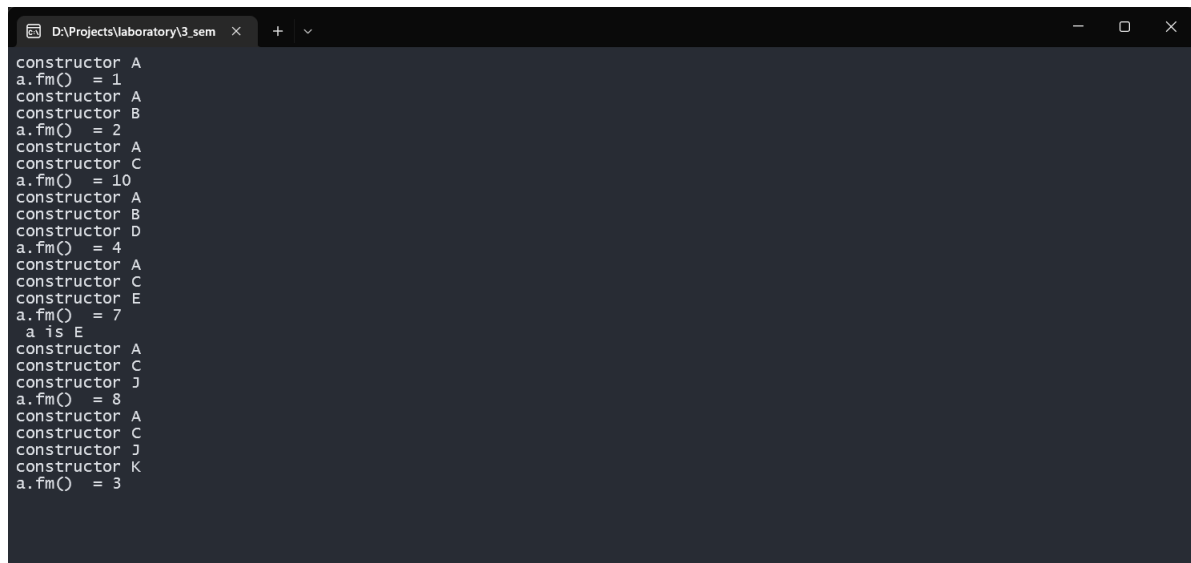
```

```

73         this.mult = 2;
74     }
75     public override int fm() {
76         return this.count + this.mult;
77     }
78 }
79
80 internal class Program
81 {
82     static void Main(string[] args)
83     {
84         A a = new A();
85         Console.WriteLine($"a.fm() = {a.fm()}");
86         a = new B();
87         Console.WriteLine($"a.fm() = {a.fm()}");
88         a = new C();
89         Console.WriteLine($"a.fm() = {a.fm()}");
90         a = new D();
91         Console.WriteLine($"a.fm() = {a.fm()}");
92         a = new E();
93         Console.WriteLine($"a.fm() = {a.fm()}");
94         if (a is E) {
95             Console.WriteLine(" a is E");
96         }
97         a = new J();
98         Console.WriteLine($"a.fm() = {a.fm()}");
99         a = new K();
100        Console.WriteLine($"a.fm() = {a.fm()}");
101
102        Console.ReadKey();
103    }
104 }
105 }

```

Результат работы



```
D:\Projects\laboratory\3_sem x + -
constructor A
a.fm() = 1
constructor A
constructor B
a.fm() = 2
constructor A
constructor C
a.fm() = 10
constructor A
constructor B
constructor D
a.fm() = 4
constructor A
constructor C
constructor E
a.fm() = 7
  a is E
constructor A
constructor C
constructor J
a.fm() = 8
constructor A
constructor C
constructor J
constructor K
a.fm() = 3
```

Вывод

По указателю супер-класса можно подставлять любой объект подкласса и работать с ним так же как и с объектом супер-класса. Замещение виртуальных функций и их вызов происходит автоматически.