

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа 1 по курсу ООП: основы программирования на языке C#

1. АГРЕГАЦИЯ ПО ССЫЛКЕ

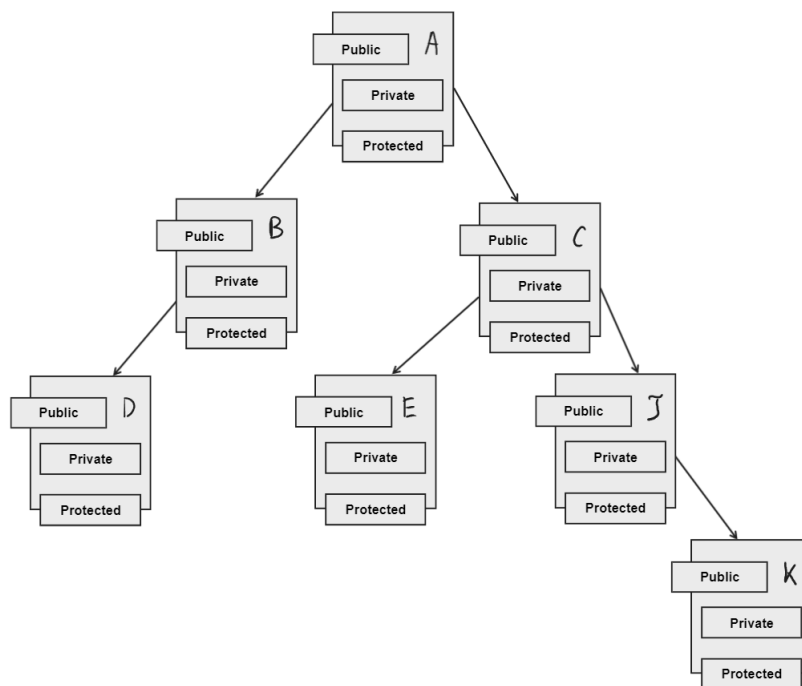
Работу выполнил:

М8О-205Б-21 Титеев Р.М. _____
(подпись) (вариант)

Руководитель: _____/Кузнецова С.В.
(подпись)

Дата: ____ октября 2022

Агрегация по ссылке



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_1
8 {
9     class A
10    {
11        private B b = null;
12        private C c = null;
13        public A(B b, C c)
14        {
15            this.b = b;
16            this.c = c;
17        }
18        public void mA()
19        {
20            Console.WriteLine("method of A");
21        }
22        public B bA
23        {
24            set { Console.WriteLine("set b"); b = value; }
25            get { Console.Write("get b ->"); return b; }
```

```

26     }
27     public C cA
28     {
29         set { Console.WriteLine("set c"); c = value; }
30         get { Console.Write("get c ->"); return c; }
31     }
32 }
33
34 class B
35 {
36     private D d = null;
37     public B(D d)
38     {
39         this.d = d;
40     }
41     public void mB()
42     {
43         Console.WriteLine(" method of B");
44     }
45     public D dA
46     {
47         set { Console.WriteLine("set d"); d = value; }
48         get { Console.Write("get d ->"); return d; }
49     }
50 }
51
52 class C
53 {
54     private J j = null;
55     private E e = null;
56     public C(J j, E e)
57     {
58         this.j = j;
59         this.e = e;
60         this.c_val = 10;
61     }
62     public void mC()
63     {
64         Console.WriteLine(" method of C");
65     }
66     public E eA
67     {
68         set { Console.WriteLine("set e"); e = value; }
69         get { Console.Write("get e ->"); return e; }
70     }
71     public J jA
72     {
73         set { Console.WriteLine("set j"); j = value; }

```

```

74         get { Console.Write("get j ->"); return j; }
75     }
76     public int c_val { set; get; }
77 }
78
79 class D
80 {
81     public D() { }
82     public void mD()
83     {
84         Console.WriteLine(" method of D");
85     }
86 }
87
88 class E
89 {
90     private D d = null;
91     public E(D d)
92     {
93         this.d = d;
94     }
95     public void mE()
96     {
97         Console.WriteLine(" method of E");
98     }
99
100     public D dA
101     {
102         set { Console.WriteLine("set d"); d = value; }
103         get { Console.Write("get d ->"); return d; }
104     }
105 }
106
107 class J
108 {
109     private K k = null;
110     public J(K k)
111     {
112         this.k = k;
113     }
114     public void mJ()
115     {
116         Console.WriteLine(" method of J");
117     }
118
119     public K kA
120     {
121         set { Console.WriteLine("set k"); k = value; }

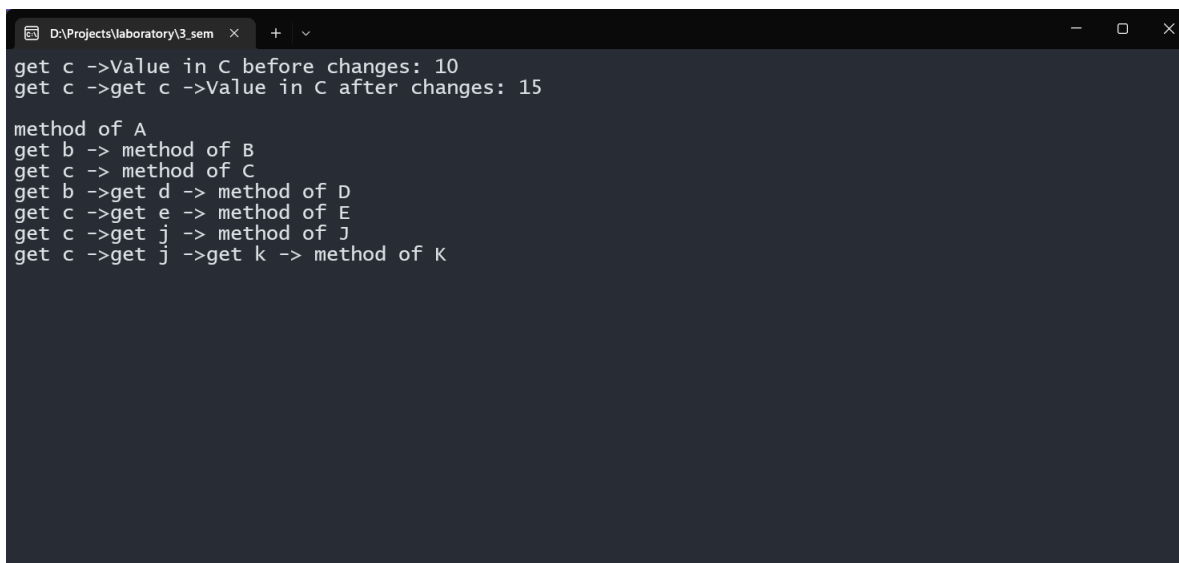
```

```

122         get { Console.Write("get k ->"); return k; }
123     }
124 }
125 class K
126 {
127     public K() { }
128     public void mK()
129     {
130         Console.WriteLine(" method of K");
131     }
132 }
133
134 internal class Program
135 {
136     static void Main(string[] args)
137     {
138         K k = new K();
139         D d = new D();
140         E e = new E(d);
141         J j = new J(k);
142
143         B b = new B(d);
144         C c = new C(j, e);
145
146         A a = new A(b, c);
147
148         Console.WriteLine($"Value in C before changes: {a.cA.c_val}");
149
150         a.cA.c_val = 15;
151
152         Console.WriteLine($"Value in C after changes: {a.cA.c_val}");
153
154         Console.WriteLine();
155         a.mA();
156         a.bA.mB();
157         a.cA.mC();
158
159         a.bA.dA.mD();
160         a.cA.eA.mE();
161         a.cA.jA.mJ();
162
163         a.cA.jA.kA.mK();
164         Console.ReadKey();
165
166     }
167 }
168
169 }

```

Результат работы



```
get c ->Value in C before changes: 10
get c ->get c ->Value in C after changes: 15

method of A
get b -> method of B
get c -> method of C
get b ->get d -> method of D
get c ->get e -> method of E
get c ->get j -> method of J
get c ->get j ->get k -> method of K
```

Пример использования

Данный метод можно использовать в качестве описания структуры работников в какой-либо компании. Так например есть главный класс и это будет главный человек в компании. Сначала мы инициализируем всех сотрудников данной компании(при этом инициализация происходит по иерархии снизу вверх). Затем мы привязываем сотрудников к своим начальникам. Кроме этого, мы можем использовать уже проинициализированного работника в других целях, и при этом его св-ва будут сохраняться в старой компании. **Вывод**

Объекты классов A, B, C, D, E, J, K существуют независимо друг от друга. При этом, например связывание объекта a с объектами b, c происходит с помощью конструктора; b, c – параметры для конструктора A. Аналогичным образом происходит связывание b с d, c с e, j и j с k. Но при этом объекты могут быть уничтожены по отдельности, что нарушает целостность структуры.