

Комбинирование через общих предков.

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_5
8 {
9     public interface C
10    {
11        void mC();
12        int fC();
13    }
14
15    public class E : C
16    {
17        private int e;
18        public E()
19        {
20            Console.WriteLine("    create E()");
21            this.e = 10;
22        }
23        public E(int e)
24        {
25            Console.WriteLine("    create E(int e) ");
26            this.e = e;
27        }
28        public int fE() { return this.e; }
29
30        public void mC() {
31            this.e = 0;
32        }
33        public int fC() { return this.e * 10; }
34    }
35
36    public interface J : C
37    {
38        int fJ();
39        void mJ();
40    }
41
42    public class K : E, J
43    {
44        private int a = 0;
```

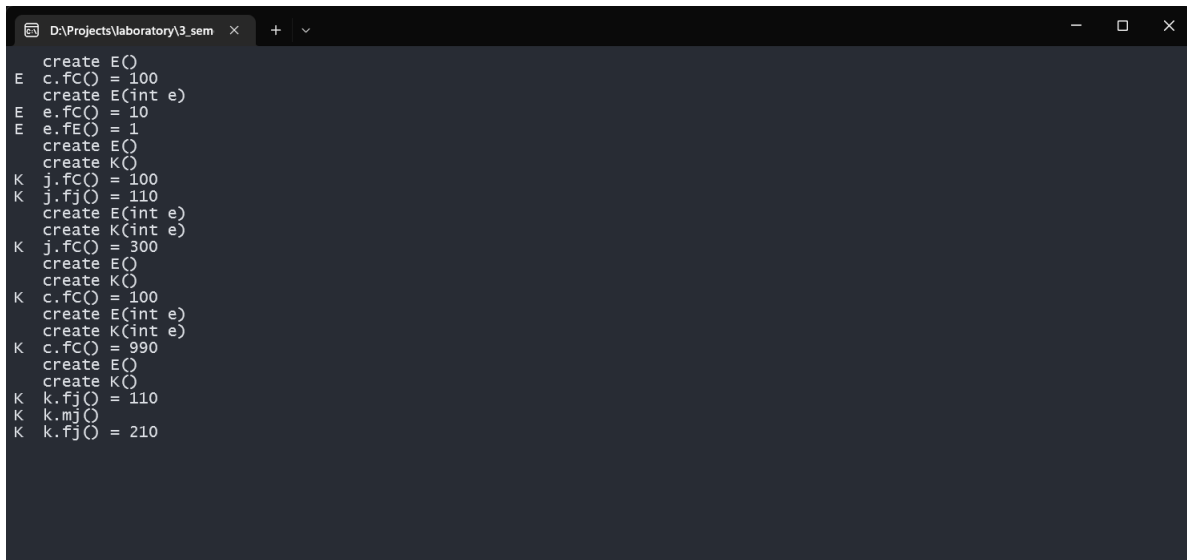
```

45     public K() {
46         Console.WriteLine("    create K() ");
47         this.a = this.fC() + this.fE();
48     }
49
50     public K(int e) : base(e)
51     {
52         Console.WriteLine("    create K(int e) ");
53         this.a = this.fE()+1;
54     }
55     public int fJ() { return this.a; }
56     public void mJ() {
57         Console.WriteLine("K  k.mj()");
58         this.a = this.fC() + this.fJ();
59     }
60 }
61
62 class Program
63 {
64     static void Main(string[] args)
65     {
66         C c = null;
67         c = new E();
68         Console.WriteLine("E  c.fC() = {0}", c.fC());
69         E e = new E(1);
70         Console.WriteLine("E  e.fC() = {0}", e.fC());
71         Console.WriteLine("E  e.fE() = {0}", e.fE());
72
73         J j = null;
74         j = new K();
75         Console.WriteLine("K  j.fC() = {0}", j.fC());
76         Console.WriteLine("K  j.fj() = {0}", j.fJ());
77         j = new K(30);
78         Console.WriteLine("K  j.fC() = {0}", j.fC());
79
80         c = new K();
81         Console.WriteLine("K  c.fC() = {0}", c.fC());
82
83         c = new K(99);
84         Console.WriteLine("K  c.fC() = {0}", c.fC());
85
86         K k = new K();
87         Console.WriteLine("K  k.fj() = {0}", k.fJ());
88         k.mJ();
89         Console.WriteLine("K  k.fj() = {0}", k.fJ());
90
91         Console.ReadKey();
92     }

```

```
93     }  
94 }
```

Результат работы



```
D:\Projects\laboratory\3_sem  x + -  
create EQ  
E c.fc() = 100  
create E(int e)  
E e.fc() = 10  
E e.fE() = 1  
create EQ  
create KO  
K j.fc() = 100  
K j.fj() = 110  
create E(int e)  
create K(int e)  
K j.fc() = 300  
create EQ  
create KO  
K c.fc() = 100  
create E(int e)  
create K(int e)  
K c.fc() = 990  
create EQ  
create KO  
K k.fj() = 110  
K k.mj()  
K k.fj() = 210
```

Вывод

Позволяет реализовать множественное наследование через общий интерфейс; комбинирует в себе интерфейс с классом с общим интерфейсом, который, в свою очередь, сам присутствует только в единственном экземпляре благодаря virtual наследованию, позволяющее избежать дублирование этого интерфейса.