

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Звіт
про виконання лабораторної роботи №5
«Множинне успадкування»

Виконав:
студент групи ФЕП-11с
Качмар Д. Б

Викладач:
Кушнір Олексій
Олександрович

Мета роботи: Проробити на практиці можливі варіанти множинного успадкування та методи уникнення проблем із ним.

Виконання роботи

Посилання на GitHub: <https://github.com/D-Kachm/OOP/tree/main/Lab5>

```
main.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int instrumentDestructorsCalled = 0;
5
6  class Instrument {
7      string material;
8      int volume;
9      string sound;
10
11 public:
12     Instrument() {
13         cout << "Instrument constructor\n";
14     }
15
16     virtual ~Instrument() {
17         instrumentDestructorsCalled++;
18         cout << "Instrument destructor called. Count: " << instrumentDestructorsCalled << endl;
19     }
20
21     int SetMaterial(string m) {
22         material = m;
23         return 1;
24     }
25
26     int SetVolume(int v) {
27         volume = v;
28         return 1;
29     }
30
31     int SetSound(string s) {
32         sound = s;
33         return 1;
34     }
35
36     int Play() {
37         cout << "something is playing..." << endl;
38         return 1;
39     }
40
41 protected:
42     int Describe() {
43         cout << "Material: " << material << ", Volume: " << volume << ", Sound: " << sound << endl;
44         return 1;
45     }
46 };
47
48 // Віртуальне успадкування для уникнення дублювання Instrument
49 class Guitar : virtual public Instrument {
50 public:
51     Guitar() {
52         cout << "Guitar constructor\n";
53         SetMaterial("wood");
54         SetVolume(70);
55         SetSound("strumming");
56         Describe();
57         Play();
58     }
59
60     virtual ~Guitar() {
61         cout << "Guitar destructor\n";
62     }
63
64     int Play() {
65         cout << "Strum-strum!" << endl;
66         return 1;
67     }
68 };
69
```

```

70 class Piano : virtual public Instrument {
71 public:
72     Piano() {
73         cout << "Piano constructor\n";
74         SetMaterial("wood and metal");
75         SetVolume(80);
76         SetSound("classical chord");
77         Describe();
78         Play();
79     }
80
81     virtual ~Piano() {
82         cout << "Piano destructor\n";
83     }
84
85     int Play() {
86         cout << "Plink-plonk!" << endl;
87         return 1;
88     }
89 };
90
91 // Множинне успадкування від Guitar і Piano
92 class MultiInstrument : public Guitar, public Piano {
93 public:
94     MultiInstrument() {
95         cout << "MultiInstrument constructor\n";
96     }
97
98     ~MultiInstrument() {
99         cout << "MultiInstrument destructor\n";
100     }
101
102     int PlayAll() {
103         Guitar::Play();
104         Piano::Play();
105         return 1;
106     }
107 };
108
109 // Успадкування від трьох класів (для демонстрації порядку)
110 class Performer : public MultiInstrument {
111     string name;
112 public:
113     Performer(string n) {
114         name = n;
115         cout << "Performer constructor: " << name << endl;
116     }
117
118     ~Performer() {
119         cout << "Performer destructor: " << name << endl;
120     }
121
122     void Show() {
123         cout << name << " is performing with multiple instruments!\n";
124         PlayAll();
125     }
126 };
127
128 int main() {
129     Performer* p = new Performer("Alex");
130     p->Show();
131     delete p;
132
133     return 0;
134 }
135

```

Результати

```
PS D:\Github\OOP\Lab5> ./main.exe
Instrument constructor
Guitar constructor
Material: wood, Volume: 70, Sound: strumming
Strum-strum!
Piano constructor
Material: wood and metal, Volume: 80, Sound: classical chord
Plink-plonk!
MultiInstrument constructor
Performer constructor: Alex
Alex is performing with multiple instruments!
Strum-strum!
Plink-plonk!
Performer destructor: Alex
MultiInstrument destructor
Piano destructor
Guitar destructor
Instrument destructor called. Count: 1
PS D:\Github\OOP\Lab5> 
```

Висновок: У цій лабораторній роботі я навчився працювати з множинним та ромбовидним успадкуванням у C++. Я зрозумів, у чому полягає проблема ромба та як її вирішити за допомогою віртуального успадкування. Також я навчився контролювати порядок виклику конструкторів і деструкторів у складних ієрархіях класів. Отримані знання допомогли мені краще зрозуміти механізми об'єктно-орієнтованого програмування та закріпити практичні навички роботи з класами.