

ГУАП

КАФЕДРА № 43

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Множественное наследование в языке C++.

по курсу: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4136

\_\_\_\_\_  
подпись, дата

Бобрович Н. С.  
\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2022

**Цель работы:**

Получение практических навыков при использовании множественного наследования в языке C++.

**Закрепить знания по теме:**

В работе необходимо построить иерархию классов согласно схеме наследования, приведенной в варианте задания.

**Условие:**

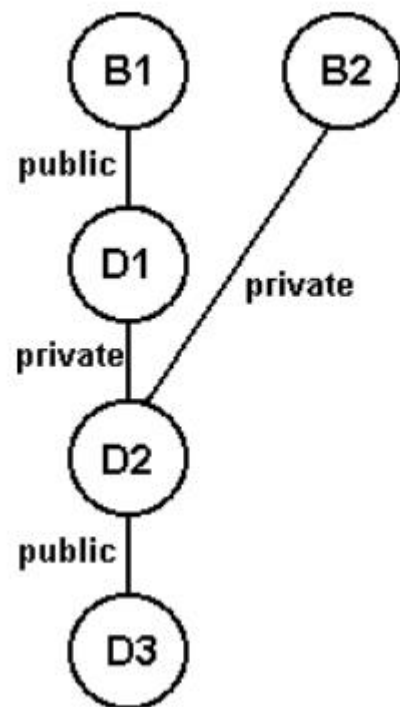
Каждый класс должен содержать:

- инициализирующий конструктор с выводом значения, полученного в качестве формального параметра.
- функцию show для вывода значений.
- деструктор.

При определении производного класса указывать базовые классы, которые являются непосредственно родительскими для данного класса. Функция main должна иллюстрировать иерархию наследования. В функции main создаем объекты производных классов ТОЛЬКО «самого нижнего» уровня в иерархии наследования.

**Вариант 9.**

9



**Листинг программы:**

```

1      #include <iostream>
2
3      using namespace std;
4
5      struct B1 {
6          string data;
7          B1() : data("B1") {}
8
9          explicit B1(const string& data) : data(data + "::B1") {}
10
11         virtual void show() {
12             cout << data << ".show()" << endl;
13         }
14     };
15
16     struct B2 {
17         string data;
18
19         B2() : data("B2") {}
20
21         explicit B2(const string& data) : data(data + "::B2") {}
22
23         virtual void show() {
24             cout << data << ".show()" << endl;
25         }
26     };
27
28     struct D1 : public B1 {
29         explicit D1(const string& data) : B1(data + "::D1") {}
30
31         D1() : B1("D1") {
32             B1::show();
33         }
34
35         void show() override {
36             B1::show();
37         }
38     };
39
40     struct D2 : private B2, private D1 {
41         explicit D2(const string& data) : B2(data + "::D2"), D1(data + "::D2") {}
42
43         D2() : B2("D2"), D1("D2") {
44             D1::show();
45             B2::show();
46         }
47
48         // поскольку в этом задании нет виртуального наследования,

```

```

49 // возникает два метода с одинаковым именем и надо решить, какой вы
50 // показано, как вызывать оба, соответственно, строки вывода будут
51 void show() override {
52     D1::show();
53     B2::show();
54 }
55 };
56
57 struct D3 : public D2 {
58     D3() : D2("D3") {
59         D2::show();
60     }
61
62     void show() override {
63         D2::show();
64     }
65 };
66
67 int main() {
68
69     system("color F0");
70     setlocale(0, "Rus");
71
72     B1 b1;
73
74     B2 b2;
75     D1 d1;
76     D2 d2;
77     D3 d3;
78
79     b1.show();
80     b2.show();
81     d1.show();
82     d2.show();
83     d3.show();
84
85     return 0;
86 }

```

**Результат работы:**

Консоль отладки Microsoft Visual Studio

```
D1::B1.show()
D2::D1::B1.show()
D2::B2.show()
D3::D2::D1::B1.show()
D3::D2::B2.show()
B1.show()
B2.show()
D1::B1.show()
D2::D1::B1.show()
D2::B2.show()
D3::D2::D1::B1.show()
D3::D2::B2.show()

C:\Users\User\source\repos\OOPLR5\x64\Debug\OOPLR5.exe (процесс 15652) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:■
```

## Вывод:

В результате выполнения работы были получены навыки обращения