

OPR Praktikum

Übung: Ausgabe

Thomas Mahr

19. März 2018

1 Lernziele

- Statische Klassenelemente verstehen
- Virtuelle Funktionen verstehen

2 Voraussetzungen

Bis Ende des Kapitels *Statische Klassenelemente* des Vorlesungsteils *Vertiefung*.

3 Aufgabe

Gegeben ist das folgende Programm:

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  class A
6  {
7  public:
8      void a() {}
9  };
10
11  class B : public A
12  {
13  public:
14      B(string s="")
15      {
16          id = ++idStat;
17          this->s = s;
18          info("B() ");
19      }
20      virtual ~B()
21      {
22          info("~B() ");
23      }
24      void a()
25      {
26          s += "1";
27          info("B::a() ");
28      }
29      virtual void b()
30      {
31          s += "2";
32          info("B::b() ");
33      }
34      void b() const
35      {
36          info("B::b() const");
37      }
38      static int liefereIdStat()
39      {
40          return idStat;
41      }
42  protected:
43      string s;
44      void info(const string&
45                text) const
46      {
47          cout << id << ": " <<
48               text << " " << s <<
```

```

47         endl;
48     }
49     private:
50         int id;
51         static int idStat;
52     };
53     int B::idStat = 0;
54
55     class C : public B
56     {
57     public:
58         C() : B("C")
59         {
60             info("C()");
61         }
62         virtual ~C()
63         {
64             info("~C()");
65         }
66         void a()
67         {
68             s += "3";
69             info("C::a()");
70         }
71         void b()
72         {
73             s += "4";
74             info("C::b()");
75         }
76     };
77
78     int main()
79     {
80         A a;
81         a.a();
82         cout << "vvv " << B::
            liefereIdStat() << endl;
83         B b;
84         b.b();
85         cout << "www " << b.
            liefereIdStat() << endl;
86         C* c[2];
87         c[0] = new C;
88         c[1] = new C;
89         c[0]->a();
90         c[1]->b();
91         delete c[0];
92         delete c[1];
93         const C c2;
94         cout << "xxx " << C::
            liefereIdStat() << endl;
95         A* ac = new C;
96         ac->a();
97         delete ac;
98         cout << "yyy " << B::
            liefereIdStat() << endl;
99         B* bc = new C;
100        bc->a();
101        bc->b();
102        delete bc;
103        cout << "zzz " << B::
            liefereIdStat() << endl;
104    }

```

Listing 1: main.cpp

Hinweis: Das Funktionsargument `s` des Konstruktors in Zeile 14 besitzt mit `s=""` eine Vorbelegung. Damit ist der Konstruktor als Standardkonstruktor nutzbar.

3.1 Programmausgabe

1. Überlegen Sie sich, was das Programm ausgegeben wird und notieren Sie sich Ihre Annahme.
2. Übersetzen Sie das Programm und führen es aus.
3. Vergleichen Sie die tatsächliche Ausgabe mit Ihrer Annahme.
4. Versuchen Sie die Unterschiede zu verstehen.
 - Tauschen Sie sich dabei mit Ihren Kommilitonen aus.
 - Befragen Sie die Tutoren.
5. Nutzen Sie den Praktikumstermin, um mit dem Übungsleiter Ihre offenen Fragen zu diskutieren.

Hinweis: Obwohl Sie bei dieser Teilaufgabe nichts Schriftliches abgeben müssen, sollten Sie sich ernsthaft und intensiv mit der Aufgabe beschäftigen, und die Gelegenheit nutzen, Ihren Kenntnisstand zu ermitteln.

3.2 Programmmodifikation

Ändern Sie das Programm so ab, dass es die unten aufgeführte Ausgabe liefert. Hierbei dürfen Sie nur folgende Änderungen vornehmen:

1. Schlüsselwort **const** hinzufügen oder entfernen
2. Schlüsselwort **virtual** hinzufügen oder entfernen
3. Destruktoren hinzufügen oder entfernen

```
vvv 0
1: B()
1: B::b() const
www 1
2: B() C
2: C() C
3: B() C
3: C() C
2: C::a() C3
3: C::b() C4
2: ~C() C3
2: ~B() C3
3: ~C() C4
3: ~B() C4
4: B() C
4: C() C
xxx 4
5: B() C
5: C() C
5: C::a() C3
5: ~C() C3
5: ~B() C3
yyy 5
6: B() C
6: C() C
6: C::a() C3
6: B::b() C32
6: ~C() C32
6: ~B() C32
zzz 6
4: ~C() C
4: ~B() C
1: ~B()
```

Einen kleinen Tipp zur Aufgabe finden Sie auf der nächsten Seite. (Umblättern auf eigene Gefahr!)

Hinweis: Sie brauchen das Programm nur an vier Stellen zu ändern.