

OPR Praktikum

Übung: Musiker

Thomas Mahr

9. Mai 2023

1 Lernziele

- Virtuelle Funktionen verwenden
- Virtuelle Destruktoren verwenden

2 Voraussetzungen

Kapitel *Späte Bindung (Polymorphie)* des zweiten Vorlesungsteils *Objektorientierte Konzepte*

3 Aufgabe

Setzen Sie die im Klassendiagramm aus Abb. 1 gezeigte Architektur in einem Orchestersimulationsprogramm um. Berücksichtigen Sie dabei diese Vorgaben:

1. Sie dürfen zwischen den gezeigten Klassen keine weiteren Abhängigkeiten herstellen als die im Klassendiagramm gezeigten.
2. Sie dürfen den Klassen nur private Datenelemente hinzufügen.
3. Das Programm fragt den Benutzer nach der Anzahl der im Orchester spielenden Musiker.
4. Jedem Musiker wird zufällig ein Musikinstrument zugeordnet.
5. Die Klänge der verschiedenen Musikinstrumente unterscheiden sich:
 - Ein Klavier klimpert.
 - Eine Trompete trötet.
 - Ein Musikinstrument spielt irgendwie.
6. Das Spielen der Musikinstrumente wird durch eine entsprechende Konsolenausgabe über die Funktion `void spielen()` simuliert.
7. Das Orchester übernimmt die Speicherverwaltung für die Musiker und gibt am Ende deren Speicherplatz frei.
8. Ein Musiker übernimmt die Speicherverwaltung für das ihm zugeordnete Musikinstrument und gibt am Ende dessen Speicherplatz frei.
9. Jede Klasse muss in einer eigenen Datei definiert sein.
10. Sie dürfen `main.cpp` nicht ändern.
11. Überprüfen Sie das Programm mit Valgrind¹ oder einem anderen Speicher-Checker.

¹www.valgrind.org

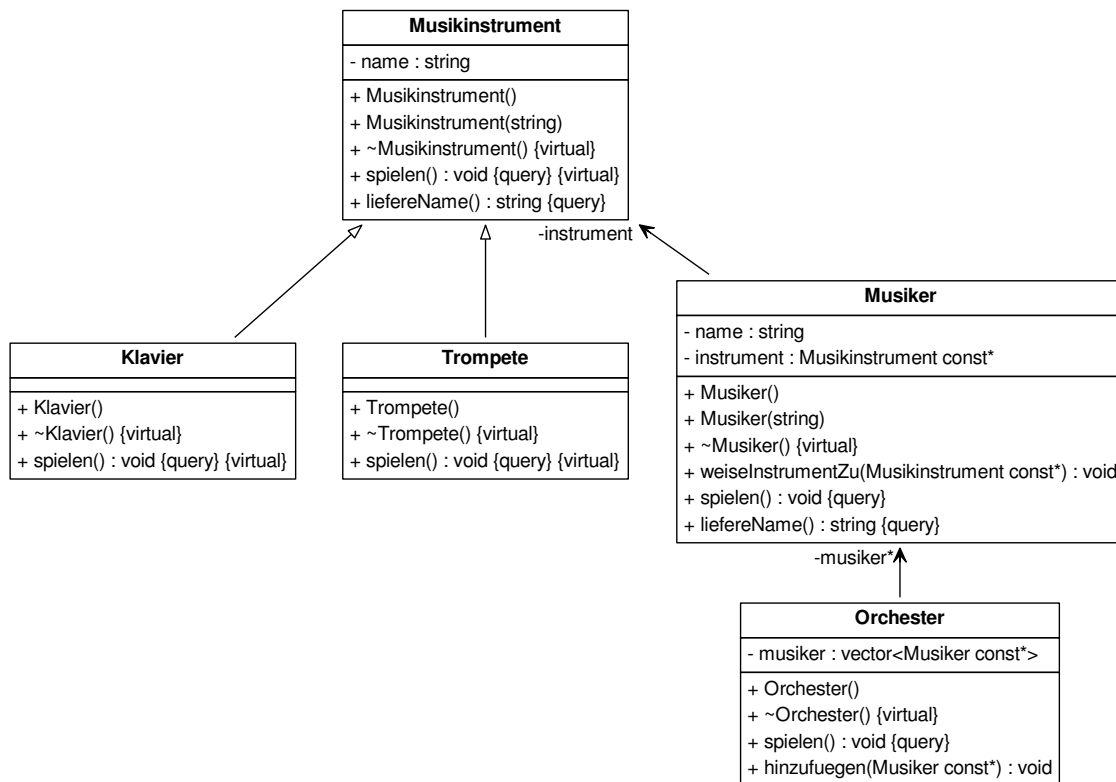


Abbildung 1: UML-Klassendiagramm.

Das Testprogramm *main.cpp* ist vorgegeben:

```

1  #include "Klavier.h"
2  #include "Trompete.h"
3  #include "Orchester.h"
4  #include <cstdlib>
5  #include <ctime>
6  #include <climits>
7  #include <iostream>
8  #include <string>
9  using namespace std;
10
11 Musikinstrument* musikinstrumentErzeugen() {
12     Musikinstrument* musikinstrument;
13     int r = rand()%3;
14     switch(r) {
15         case 0:
16             musikinstrument = new Klavier();
17             break;
18         case 1:
19             musikinstrument = new Trompete();
20             break;
21         default:
22             musikinstrument = new Musikinstrument();
23             break;
  
```

```

24     }
25     return musikinstrument;
26 }
27
28 void test1() {
29     cout << "*** test1 ***\n";
30     Musiker* musiker = new Musiker("Musiker 1");
31     musiker->spielen();
32     delete musiker;
33     musiker = new Musiker("Musiker 2");
34     musiker->weiseInstrumentZu(new Klavier());
35     musiker->weiseInstrumentZu(new Trompete());
36     musiker->spielen();
37     delete musiker;
38 }
39
40 void test2() {
41     cout << "*** test2 ***\n";
42     srand(time(0));
43     Orchester orchester;
44     int n;
45     cout << "Wieviele Musiker sollen im Orchester spielen? ";
46     cin >> n;
47     cin.ignore(INT_MAX, '\n');
48     cout << "+++ Orchester mit " << n << " Musikern besetzen\n";
49     for(int i=0; i<n; i++)
50     {
51         cout << "Wie heit der " << i+1 << ". Musiker? ";
52         string name;
53         getline(cin, name);
54         Musiker* musiker = new Musiker(name);
55         orchester.hinzufuegen(musiker);
56         musiker->weiseInstrumentZu(musikinstrumentErzeugen());
57     }
58     cout << "+++ Das Konzert beginnt\n";
59     orchester.spielen();
60     cout << "+++ Applaus\n";
61 }
62
63 int main() {
64     test1();
65     test2();
66 }

```

Listing 1: main.cpp

Eine exemplarische Konsolenausgabe knnte so aussehen:

```

*** test1 ***
Konstruktor: Musiker(string name): Musiker 1
Destruktor: ~Musiker(): Musiker 1
Konstruktor: Musiker(string name): Musiker 2
Konstruktor: Musikinstrument(string name): Klavier
Konstruktor: Klavier()

```

```

Musiker 2 erhält Klavier.
Konstruktor: Musikinstrument(string name): Trompete
Konstruktor: Trompete()
Destruktor: ~Klavier()
Destruktor: ~Musikinstrument(): Klavier
Musiker 2 erhält Trompete.
Musiker 2 spielt Trompete:
Trompete trötet
Destruktor: ~Musiker(): Musiker 2
Destruktor: ~Trompete()
Destruktor: ~Musikinstrument(): Trompete
*** test2 ***
Konstruktor: Orchester()
Wieviele Musiker sollen im Orchester spielen? 3
+++ Orchester mit 3 Musikern besetzen
Wie heißt der 1. Musiker? Donald Duck
Konstruktor: Musiker(string name): Donald Duck
Konstruktor: Musikinstrument(string name): Trompete
Konstruktor: Trompete()
Donald Duck erhält Trompete.
Wie heißt der 2. Musiker? Gustav Gans
Konstruktor: Musiker(string name): Gustav Gans
Konstruktor: Musikinstrument()
Gustav Gans erhält Irgendein Instrument.
Wie heißt der 3. Musiker? Daniel Düsentrieb
Konstruktor: Musiker(string name): Daniel Düsentrieb
Konstruktor: Musikinstrument(string name): Klavier
Konstruktor: Klavier()
Daniel Düsentrieb erhält Klavier.
+++ Das Konzert beginnt
Donald Duck spielt Trompete:
Trompete trötet
Gustav Gans spielt Irgendein Instrument:
Irgendein Instrument erklingt irgendwie
Daniel Düsentrieb spielt Klavier:
Klavier klimpert
+++ Applaus
Destruktor: ~Orchester()
Destruktor: ~Musiker(): Donald Duck
Destruktor: ~Trompete()
Destruktor: ~Musikinstrument(): Trompete
Destruktor: ~Musiker(): Gustav Gans
Destruktor: ~Musikinstrument(): Irgendein Instrument
Destruktor: ~Musiker(): Daniel Düsentrieb
Destruktor: ~Klavier()
Destruktor: ~Musikinstrument(): Klavier

```