

OPR Praktikum

Übung: Musiker

Thomas Malm

6. April 2019

1 Lernziele

- Virtuelle Funktionen verwenden
- Virtuelle Destruktoren verwenden
- Dynamische Speicherverwaltung 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.

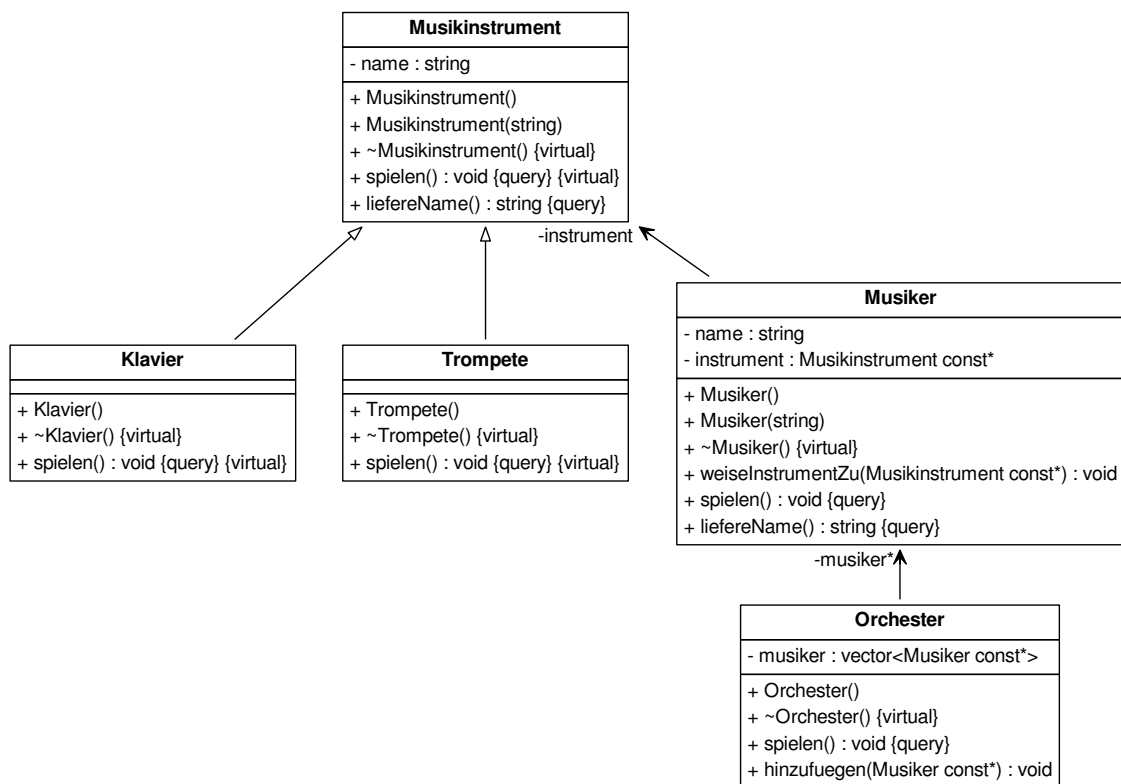


Abbildung 1: UML-Klassendiagramm.

10. Sie dürfen *main.cpp* nicht ändern.

Das Testprogramm *main.cpp* ist vorgegeben:

```
8  #include "Klavier.h"
9  #include "Trompete.h"
10 #include "Orchester.h"
11 #include <cstdlib>
12 #include <ctime>
13 #include <climits>
14 #include <iostream>
15 #include <string>
16 using namespace std;
17
18 Musikinstrument* musikinstrumentZufaelligErzeugen()
19 {
20     Musikinstrument* musikinstrument;
21     int r = rand()%3;
22     switch(r)
23     {
24         case 0:
25             musikinstrument = new Klavier();
26             break;
27         case 1:
28             musikinstrument = new Trompete();
29             break;
30         default:
31             musikinstrument = new Musikinstrument();
32             break;
33     }
34     return musikinstrument;
35 }
36
37 int main()
38 {
39     srand(time(0));
40
41     Orchester orchester;
42
43     int n;
44     cout << "Wieviele Musiker sollen im Orchester spielen? ";
45     cin >> n;
46     cin.ignore(INT_MAX, '\n'); // entfernt übriggebliebenen
47                               // Zeilenumbruch der letzten Eingabe
48
49     cout << "+++ Orchester mit " << n << " Musikern besetzen\n";
50     for(int i=0; i<n; i++)
51     {
52         cout << "Wie heißt der " << i+1 << ". Musiker? ";
53         string name;
54         getline(cin, name);
55         Musiker* musiker = new Musiker(name);
56         orchester.hinzufuegen(musiker);
57         musiker->weiseInstrumentZu(musikinstrumentZufaelligErzeugen())
58         ;
59     }
60 }
```

```
57     }  
58  
59     cout << "+++ Das Konzert beginnt\n";  
60     orchester.spielen();  
61  
62     cout << "+++ Applaus\n";  
63 }
```

Listing 1: main.cpp

Eine exemplarische Konsolenausgabe könnte so aussehen:

```

Konstruktor: Orchester()
Wieviele Musiker sollen im Orchester spielen? 5
+++ Orchester mit 5 Musikern besetzen
Wie heißt der 1. Musiker? Donald Duck
Konstruktor: Musiker(string name): Donald Duck
Konstruktor: Musikinstrument(string name): Klavier
Konstruktor: Klavier()
Donald Duck erhält Klavier
Wie heißt der 2. Musiker? Daisy Duck
Konstruktor: Musiker(string name): Daisy Duck
Konstruktor: Musikinstrument(string name): Trompete
Konstruktor: Trompete()
Daisy Duck erhält Trompete
Wie heißt der 3. Musiker? Daniel Düsentrieb
Konstruktor: Musiker(string name): Daniel Düsentrieb
Konstruktor: Musikinstrument()
Daniel Düsentrieb erhält Irgendein Instrument
Wie heißt der 4. Musiker? Gustav Gans
Konstruktor: Musiker(string name): Gustav Gans
Konstruktor: Musikinstrument(string name): Trompete
Konstruktor: Trompete()
Gustav Gans erhält Trompete
Wie heißt der 5. Musiker? Helferlein
Konstruktor: Musiker(string name): Helferlein
Konstruktor: Musikinstrument()
Helferlein erhält Irgendein Instrument
+++ Das Konzert beginnt
Donald Duck spielt Klavier:
Klavier klimpert
Daisy Duck spielt Trompete:
Trompete trötet
Daniel Düsentrieb spielt Irgendein Instrument:
Irgendein Musikinstrument erklingt irgendwie
Gustav Gans spielt Trompete:
Trompete trötet
Helferlein spielt Irgendein Instrument:
Irgendein Musikinstrument erklingt irgendwie
+++ Applaus
Destruktor: ~Orchester()
Destruktor: ~Musiker(): Donald Duck
Destruktor: ~Klavier()
Destruktor: ~Musikinstrument(): Klavier
Destruktor: ~Musiker(): Daisy Duck
Destruktor: ~Trompete()
Destruktor: ~Musikinstrument(): Trompete
Destruktor: ~Musiker(): Daniel Düsentrieb
Destruktor: ~Musikinstrument(): Irgendein Instrument
Destruktor: ~Musiker(): Gustav Gans
Destruktor: ~Trompete()
Destruktor: ~Musikinstrument(): Trompete
Destruktor: ~Musiker(): Helferlein
Destruktor: ~Musikinstrument(): Irgendein Instrument

```