Autoren: Marius Birk
Pieter Vogt
Tutor: Florian Brandt

Abgabe: 05.05.2020, 12:00 Uhr

Smileys:

A1	A2	A3	$\sum$

Objektorientierte Modellierung und Programmierung

Abgabe Uebungsblatt Nr.02

(Alle allgemeinen Definitionen aus der Vorlesung haben in diesem Dokument bestand, es sei den sie erhalten eine explizit andere Definition.)

1 Aufgabe 1

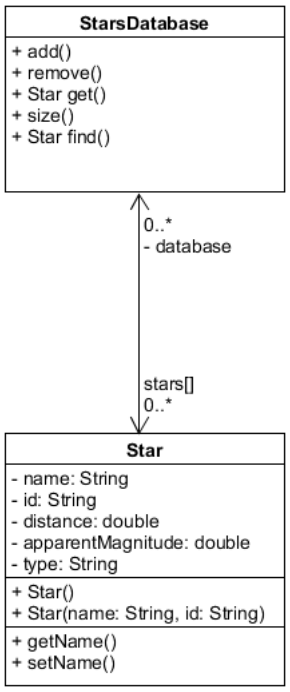


Abbildung 1: Klassendiagramm der Sternendatenbank

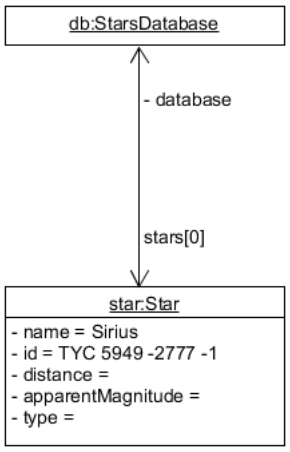


Abbildung 2: Objektdiagramm: Test 1

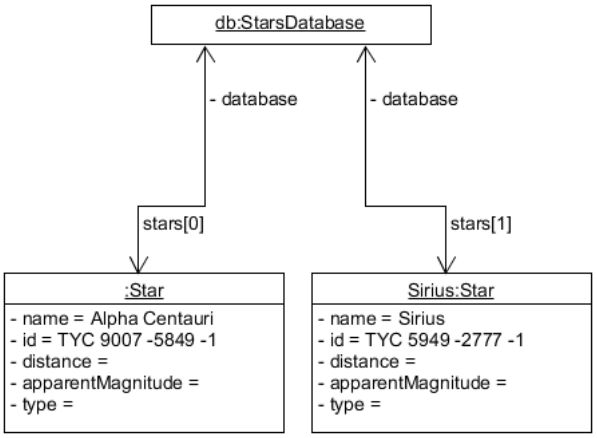


Abbildung 3: Objektdiagramm: Test 2

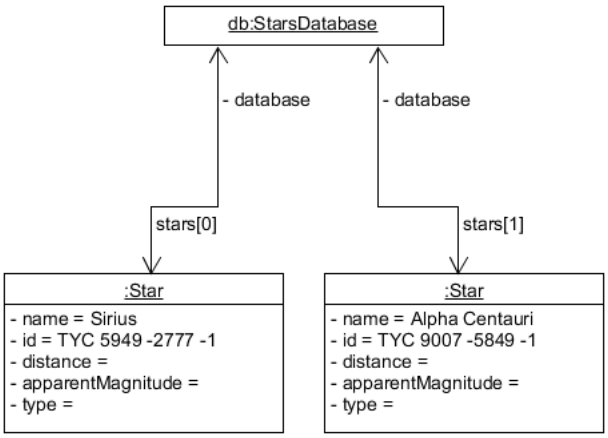
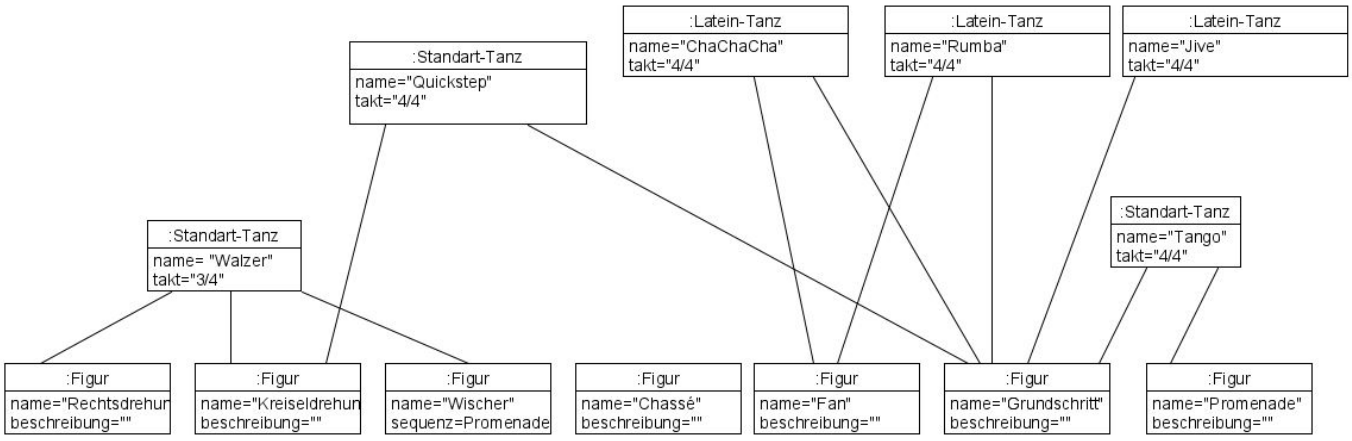


Abbildung 4: Objektdiagramm: Test 3

2 Aufgabe 2

2.1 a)



2.2 b)

2.2.1 Dance Code

```
import java.util.ArrayList;
class Dance {
private String name;
private String beat;
private Object[] figures = new Object[3];

public Object[] getFigures() {
return figures;
}
```

```
public void setFigures(Object[] figures) {
for(int i = 0; i< figures.length; i++){
this.figures[i] = figures[i];
}
}

public String getName() {
return name;
}

public void setName(String name) {
this.name = name;
}

public String getBeat() {
return beat;
}

public void setBeat(String beat) {
this.beat = beat;
}
}
class StandardDance extends Dance{

}
class LatinDance extends Dance{

}
class Figure{
private String name;
private String text;

public String getName() {
return name;
}

public void setName(String name) {
this.name = name;
}

public String getText() {
return text;
}

public void setText(String text) {
```

```
this.text = text;
}
}
class Sequence extends Figure{
private String name;
public ArrayList<Object> figures = new ArrayList<Object>();
public void setSequence(ArrayList sequence){
this.figures = sequence;
}
public boolean add(Figure figure){
if( figure instanceof Sequence){
return false;
}
else{
figures.add(figure);
return true;
}
}
}
```

2.2.2 Dance Database

```
import java.io.FileReader;
import java.lang.reflect.Array;
import java.util.ArrayList;

public class DanceDatabase {
    public static void main(String[] args) {
        Figure Righturn = new Figure();
        Righturn.setName("Rightturn");
        Righturn.setText("Turning right");

        Figure Circle = new Figure();
        Circle.setName("Circle");
        Circle.setText("Circle");

        Figure Whisk = new Figure();
        Whisk.setName("Whisk");

        Figure Chasse = new Figure();
        Chasse.setName("Chassé");
        Chasse.setText("Chassé");

        Figure Fan = new Figure();
        Fan.setText("Fan");
        Fan.setName("Fan");
```

```
Figure Basic = new Figure();
Basic.setName("Basic Movement");
Basic.setText("Basic Movement");

Figure Promenade = new Figure();
Promenade.setText("Promenade");
Promenade.setName("Promenade");

Sequence S_Whisk = new Sequence();
if(S_Whisk.add(Chasse)){
    System.out.println("Hinzufügen erfolgreich");
    //Um diese Prüfung zu realisieren musste das Objekt Array auf eine Obje
    //gespeichert werden.
}
else{
    System.out.println("Hinzufügen nicht erfolgreich");

}
StandardDance Walzer = new StandardDance();
Walzer.setName("Walzer");
Walzer.setBeat("3/4");
Object[] figures = new Object[]{Righturn, Circle, Whisk};
Walzer.setFigures(figures);
figures = null;

StandardDance Tango = new StandardDance();
Tango.setBeat("4/4");
Tango.setName("Tango");
figures= new Object[]{Basic, Promenade};
Tango.setFigures(figures);
figures= null;

StandardDance Quickstep = new StandardDance();
Tango.setBeat("4/4");
Tango.setName("Quickstep");
figures = new Object[]{Basic, Circle};
Tango.setFigures(figures);
figures= null;

LatinDance Cha = new LatinDance();
Cha.setBeat("4/4");
Cha.setName("ChaChaCha");
figures = new Object[]{Basic, Fan};
Cha.setFigures(figures);
figures= null;
```

```
        LatinDance Rumba = new LatinDance();
        Rumba.setBeat("4/4");
        Rumba.setName("Rumba");
        figures = new Object[]{Basic, Fan};
        Cha.setFigures(figures);
        figures= null;

        LatinDance Jive = new LatinDance();
        Jive.setBeat("4/4");
        Jive.setName("ChaChaCha");
        figures = new Object[]{Basic};
        Jive.setFigures(figures);
        figures= null;
    }
}
```

2.3 c)

```
public boolean add(Figure figure){
        if( figure instanceof Sequence){
            return false;
        }
        else{
            figures.add(figure);
            return true;
        }
    }
```

3 Aufgabe 3

3.1 a)

```
1  public class Out {
2
3      public static void out(boolean bool) {
4          System.out.println(bool);
5      }
6
7      public static void out(int number) {
8          System.out.println(number);
9      }
10
11     public static void out(double number) {
12         System.out.println(number);
```

```
13      }
14
15      public static void out(char character) {
16          System.out.println(character);
17      }
18
19      public static void out(String string) {
20          System.out.println(string);
21      }
22
23      public static void out(Object obj) {
24          System.out.println(obj);
25      }
26
27  }
```