

Autoren: Marius Birk
 Pieter Vogt
 Tutor: Florian Brandt

Abgabe: 05.05.2020, 12:00 Uhr

Smileys:

A1	A2	A3	Σ

Objektorientierte Modellierung und Programmierung

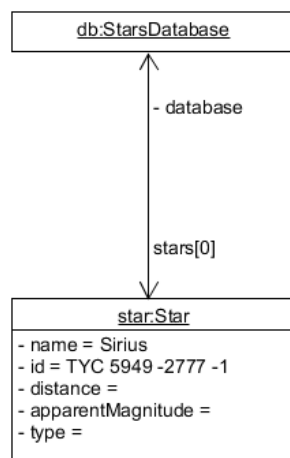
Abgabe Uebungsblatt Nr.02

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

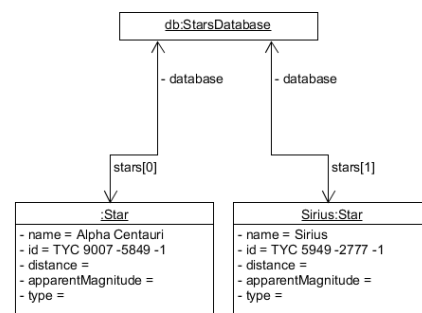
1 Aufgabe 1



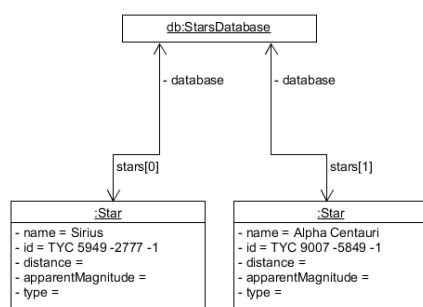
(a) Klassendiagramm Star, StarsDatabase



(b) Objektdiagramm Sirius



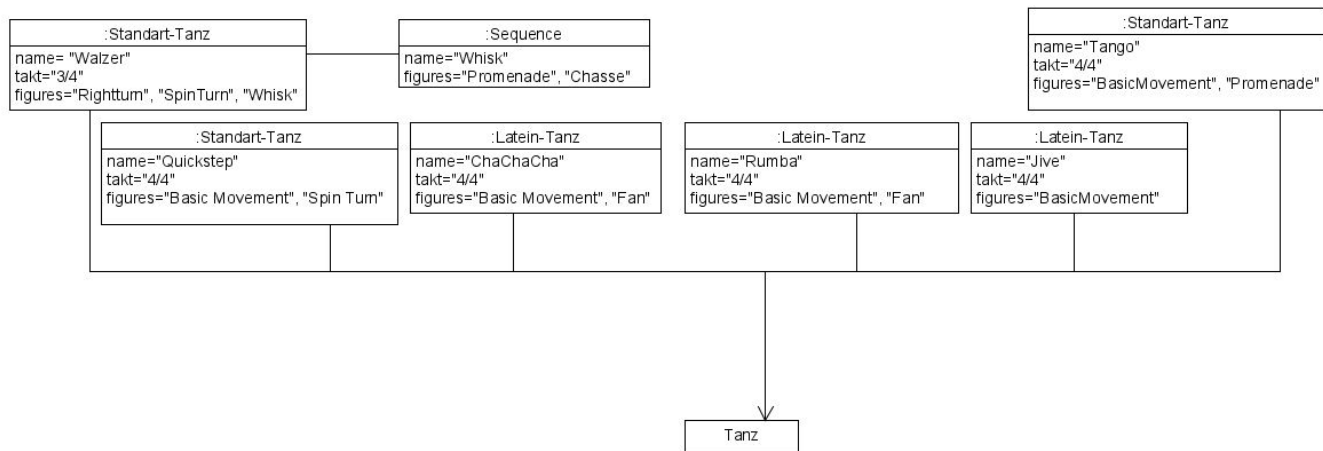
(c) Objektdiagramm Sirius, Alpha Centauri



(d) Objektdiagramm Alpha Centauri, Sirius

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 ArrayList figures = new ArrayList();

    public void getFigures() {
        System.out.println(this.figures);
    }

    public void setFigures(String Figures) {
        this.figures.add(Figures);
    }

    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 {
    public ArrayList<String> figures = new ArrayList<String>();

    public void setSequence(ArrayList sequence) {
        this.figures = sequence;
    }
}
```

```
}

public boolean add(String figure) {
    if (figures.contains(figure)) {
        return false;
    } else {
        figures.add(figure);
        return true;
    }
}
}
```

2.2.2 Dance Database

```
import java.io.FileReader;
import java.util.ArrayList;

public class DanceDatabase {
    public static void main(String[] args) {
        ArrayList<String> Figures = new ArrayList<String>();
        Figures.add("Rightturn");
        Figures.add("Basic Movement");
        Figures.add("Spin Turn");
        Figures.add("Promenade");
        Figures.add("Chasse");
        Figures.add("Fan");
        Figures.add("Whisk");

        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 Object ArrayList
            //gespeichert werden.
        } else {
            System.out.println("Hinzufügen nicht erfolgreich");
        }
        if (S_Whisk.add("Promenade")) {
            System.out.println("Hinzufügen erfolgreich");
            //Um diese Prüfung zu realisieren musste das Objekt Array auf eine Object ArrayList
            //gespeichert werden.
        } else {
            System.out.println("Hinzufügen nicht erfolgreich");
        }
        StandardDance Walzer = new StandardDance();
        Walzer.setName("Walzer");
        Walzer.setBeat("3/4");
```

```
Walzer.setFigures(Figures.get(0));
Walzer.setFigures(Figures.get(2));
Walzer.setFigures(Figures.get(6));

StandardDance Tango = new StandardDance();
Tango.setBeat("4/4");
Tango.setName("Tango");
Tango.setFigures(Figures.get(1));
Tango.setFigures(Figures.get(3));

StandardDance Quickstep = new StandardDance();
Quickstep.setBeat("4/4");
Quickstep.setName("Quickstep");
Quickstep.setFigures(Figures.get(1));
Quickstep.setFigures(Figures.get(2));

LatinDance Cha = new LatinDance();
Cha.setBeat("4/4");
Cha.setName("ChaChaCha");
Cha.setFigures(Figures.get(1));
Cha.setFigures(Figures.get(5));

LatinDance Rumba = new LatinDance();
Rumba.setBeat("4/4");
Rumba.setName("Rumba");
Rumba.setFigures(Figures.get(1));
Rumba.setFigures(Figures.get(5));

LatinDance Jive = new LatinDance();
Jive.setBeat("4/4");
Jive.setName("ChaChaCha");
Jive.setFigures(Figures.get(1));
}

}
```

2.3 c)

```
public boolean add(String figure) {
    if (figures.contains(figure)) {
        return false;
    } else {
        figures.add(figure);
        return true;
    }
}
```

}

3 Aufgabe 3

3.1 a)

```
public class Out {  
    public void out(String s){  
        System.out.println(s);  
    }  
    public void out(boolean b){  
        System.out.println(b);  
    }  
    public void out(int i){  
        System.out.println(i);  
    }  
    public void out(double d){  
        System.out.println(d);  
    }  
    public void out(char c){  
        System.out.println(c);  
    }  
  
    public void out(Object o){  
        System.out.println(o);  
    }  
}
```