

Übungsblatt 8

Abgabe bis Freitag, 21.12.2012, 12:00 Uhr

Hinweis: Lösungen immer per Email an den zuständigen Tutor schicken. Die Emailadressen sind auf der Homepage zur Vorlesung gelistet. Verwenden Sie für alle Programmieraufgaben die Struktur des Beispielprojekts ¹. Kompilieren Sie Ihre Projekte mit Hilfe des ant-Buildsystems.

Aufgabe 8.1

In dieser Aufgabe sollen Schiffe modelliert werden. Dabei gilt:

- Zu jedem Schiff gehört eine Typbezeichnung.
- Zu jedem Frachtschiff gehören die Angaben Baujahr und Ladegewicht.
- Zu jedem Containerschiff gehören die Angaben Baujahr, Ladegewicht und Anzahl der Container.
- Zu jedem Tankschiff gehören die Angaben Baujahr, Ladegewicht und Tankvolumen.
- Zu jedem Kreuzfahrtschiff gehören die Angaben Baujahr, Anzahl der Kabinen.
- Eine Methode `info()` soll von jedem Objekt den Namen und zusätzlich die spezifischen Daten des Objektes als String zurückgeben.
- Von der Klasse Schiff sollen keine Objekte erzeugt werden können.

Modellieren Sie die Schiffe in Java und vermeiden Sie dabei unnötige Wiederholungen von Methoden oder Instanzvariablen.

1. Visualisieren Sie die Klassenhierarchie.
2. Implementieren Sie alle benötigten Klassen.
3. Erzeugen Sie von jedem Schiffstyp jeweils eine Instanz und speichern Sie diese in einer `ArrayList<Schiff>`. Iterieren Sie über die `ArrayList` um die spezifischen Daten aller Schiffe auszugeben.

¹<http://ais/teaching/ws12/info/material/MyProject.zip>

Aufgabe 8.2

1. Was bedeutet Polymorphismus?
2. Welche Ausgabe liefert die main-Methode der Klasse MyTest?

```
class A {
    public A() {
        System.out.println("Konstruktor A");
    }
    public void p() {
        System.out.println("A");
    }
}

class B extends A {
    public B() {
        System.out.println("Konstruktor B");
    }
}

class C extends B {
    public void p() {
        System.out.println("C");
    }
}

public class MyTest {
    public static void main(String[] args) {
        A a = new B();
        A b = new C();
        B c = new C();
        A d = (A)c;
        a.p();
        b.p();
        c.p();
        d.p();
    }
}
```

3. Die Folge der Fibonacci-Zahlen ist folgendermaßen definiert:

$$\begin{aligned}f_0 &= 0 \\f_1 &= 1 \\&\vdots \\f_n &= f_{n-1} + f_{n-2}\end{aligned}$$

- (a) Implementieren Sie eine rekursive Java-Methode, die die n -te Fibonacci-Zahl berechnet.
- (b) Schreiben Sie eine Testmethode mittels JUnit, die testet, ob Ihre Methode die zehnte und die zwölfte Fibonacci-Zahl korrekt berechnet.