

## Übungsblatt 12

Abgabe bis Montag, 24.07.2015, 10:00 Uhr

### Hinweis:

Aufgaben immer per E-Mail (eine E-Mail pro Blatt und Gruppe) an den zuständigen Tutor schicken (Bei Programmieraufgaben Java Quellcode und evtl. benötigte Datendateien).

### Aufgabe 12.1

1. Implementieren Sie ein Interface `GeometricObject`, das die Methoden `double computeArea()` und `double computeCircumference()` besitzt.
2. Vervollständigen Sie die Klasse `Rectangle` so, dass sie ebenfalls das Interface `GeometricObject` implementiert.

```
class Rectangle ... {  
    public Rectangle(double length, double width) { ... }  
  
    public double getArea() { ... }  
    public double getCircumference() { ... }  
  
    protected double length;  
    protected double width;  
}
```

3. Implementieren Sie eine Klasse `Square`, die von `Rectangle` abgeleitet ist. Die Seitenlänge des Quadrates soll im Konstruktor gesetzt werden.
4. Könnten die Instanzvariablen `length` und `width` auch als `private` deklariert werden?
5. Implementieren Sie die Klasse `GeometricObjectList`, die mehrere geometrische Objekte verwalten soll. Diese Klasse soll als Instanzvariable eine `ArrayList` `ArrayList<GeometricObject>` besitzen und folgende Methoden zur Verfügung stellen:
  - (a) `void add(GeometricObject o)`  
fügt ein Objekt in die `ArrayList` ein.
  - (b) `double areaSum()`  
berechnet die Summe des Flächeninhalts aller Objekte in der `ArrayList`.
  - (c) `double circumferenceSum()`  
berechnet die Summe des Umfangs aller Objekte in der `ArrayList`.

## Aufgabe 12.2

Für einen Fuhrpark bestehend aus PKWs, LKWs, Bussen und Fahrrädern soll eine Klassenhierarchie entworfen werden. Verwenden Sie die folgenden Klassen:

```
Fahrzeug
Kraftfahrzeug
Bus
Fahrrad
PKW
LKW
```

Die unterschiedlichen Fahrzeuge besitzen sowohl gemeinsame als auch unterschiedliche Attribute:

- Jedes Fahrzeug besitzt eine Seriennummer.
- Jedes Kraftfahrzeug besitzt einen TÜV-Termin.
- Zu jedem Bus gehören die Angaben: Baujahr, amtliche Kennzeichen, Anzahl Sitzplätze, Anzahl Stehplätze sowie die Leistung (ganzzahlig).
- Zu jedem Fahrrad gehören die Angaben: Baujahr und Rahmengröße.
- Zu jedem PKW gehören die Angaben: Baujahr, amtliche Kennzeichen, Anzahl Sitzplätze sowie die Leistung (ganzzahlig).
- Zu jedem LKW gehören die Angaben: Baujahr, amtliche Kennzeichen, Anzahl Sitzplätze, Leistung (ganzzahlig) sowie Zuladung (ganzzahlig).

Darüber hinaus soll die `toString` Methode von jedem Objekt den Typ und zusätzlich die spezifischen Daten des Objektes ausgeben.

1. Implementieren Sie eine Klassenhierarchie. Machen Sie dabei Gebrauch von Vererbung, abstrakten Klassen und Methoden. Vermeiden Sie dabei Wiederholungen.
2. Testen Sie Ihre Implementierung anhand der Klasse `TestHierarchy`, die Sie auf der Homepage zur Übung finden.
3. Visualisieren Sie Ihre Klassenhierarchie als Graphen. Zeichnen Sie ein Rechteck für jede Klasse und einen Pfeil für jede Vererbung (jeweils von der Subklasse zur Superklasse).

### Aufgabe 12.3

Betrachten Sie die folgende Klassenhierarchie. Geben Sie an, was in der `main`-Methode ausgegeben wird.

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

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

class C extends A {
    public C () { }
}

class D extends B {
    public D () { }
}

class Hierarchy {
    public static void main (String args[]) {
        A a = new C();
        A b = new B();
        B c = new B();
        A d = new D();
        D e = new D();

        a.print();
        b.print();
        c.print();
        d.print();
        e.print();
    }
}
```