

Übungsblatt 8

Abgabe bis Donnerstag, 22.12.11, 12:00 Uhr

Hinweis: Kompilieren und testen Sie Ihre Programme mit Hilfe des `ant`-Buildsystemes. Schicken Sie Ihrem Tutor eine Kopie des kompletten Projektordners, der alle benötigten Dateien enthält.

Aufgabe 8.1

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 und ein Kennzeichen.
- Zu jedem Bus gehören Anzahl der Sitzplätze und die Anzahl der Stehplätze.
- Zu jedem Fahrrad gehören die Angaben: Baujahr und Rahmengröße.
- Zu jedem PKW gehören die Angaben: Anzahl Sitzplätze und Leistung (ganzzahlig).
- Zu jedem LKW gehören die Angaben: 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. 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).
2. Implementieren Sie eine Klassenhierarchie. Machen Sie dabei Gebrauch von Vererbung, abstrakten Klassen und Methoden. Vermeiden Sie dabei Wiederholungen.

Aufgabe 8.2

1. Implementieren Sie ein Interface `GeometricObject`, das die Methoden `double computeArea()` und `double computeCircumference()` besitzt.
2. Ändern Sie die Klasse `Triangle` von Übungsblatt 4 (siehe Vorlesungshomepage) so ab, dass sie das Interface `GeometricObject` implementiert.
3. Vervollständigen Sie folgende Klasse `Rectangle`, die 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;
}
```

4. Implementieren Sie eine Klasse `Square`, die von `Rectangle` abgeleitet ist. Die Seitenlänge des Quadrates soll im Konstruktor gesetzt werden.
5. Könnten die Instanzvariablen `length` und `width` auch als `private` deklariert werden?
6. Implementieren Sie die Klasse `GeometricObjectVector`, die mehrere geometrische Objekte verwalten soll. Diese Klasse soll als Instanzvariable einen Vektor `Vector<GeometricObject>` besitzen und folgende Methoden zur Verfügung stellen:
 - (a) `void add(GeometricObject o)`
fügt ein Objekt in den Vektor ein.
 - (b) `double areaSum()`
berechnet die Summe des Flächeninhalts aller Objekte in dem Vektor.
 - (c) `double circumferenceSum()`
berechnet die Summe des Umfangs aller Objekte in dem Vektor.
 - (d) `GeometricObject maxArea()`
gibt das Objekt mit dem größten Flächeninhalt zurück.