

zu Kapitel 8

Aufgabe 1

(KfzVerwaltung.java)

Für die Kfz-Datenbank einer Behörde sollen in einer ersten Ausbaustufe nur Pkws und Lkws betrachtet werden. Für beide Kfz-Typen soll das Kennzeichen und die PS-Zahl gespeichert werden, für Pkws zusätzlich die Anzahl der Sitzplätze, für Lkws zusätzlich das zulässige Gesamtgewicht. (Später werden noch viele weitere Daten, sowie Motorräder etc. hinzukommen.) Definieren Sie geeignete Klassen Kfz, Pkw und Lkw, jeweils mit einer print()-Funktion zur Ausgabe der Daten. Als Testprogramm für Ihre Klassen legen Sie in der main()-Funktion einen Pkw und einen Lkw an und geben deren Werte aus.

Aufgabe 2

(StackTest.java)

Schreiben Sie die Klasse Stack aus der Vorlesung so um, dass beliebige Objekte (sic!) auf dem Stack gespeichert werden können. Für ein Testprogramm legen Sie wieder Bücher auf den Stack, dabei empfiehlt es sich in der Klasse Buch die print()-Funktion durch die allgemeinere Funktion toString() zu ersetzen.

Hinweis: An vielen Stellen früherer Übungsaufgaben war eine print()-Funktion gefordert, da wir toString() noch nicht kannten. toString() ist i.d.R. die bessere Wahl.

Aufgabe 3

(Zeichenprogramm.java)

- a) Für ein CAD-Programm sollen verschiedene Figuren programmiert werden (Kreise, Rechtecke, Quadrate usw.). Alle Figuren haben eine Farbe (bei uns nur ein String) und entsprechend eine setFarbe() und getFarbe()-Methode. Darüber hinaus kann von allen Figuren der Umfang und die Fläche berechnet werden. (getUmfang() und getFlaeche()). Die für die spezielle Figurenform nötigen Parameter (z.B. Radius oder Kantenlänge) können pro Figurenform ebenfalls mittels geeigneter set- und get-Funktionen gesetzt und gelesen werden. Mangels eines geeigneten Zeichenprogramms gibt die print()-Funktion alle Merkmale einer Figur aus (z.B. Typ, Farbe, Parameter, Umfang, Fläche). Programmieren Sie eine abstrakte Klasse Figur, in die Sie so viel, wie möglich gemeinsame Funktionalität aller Figuren packen. Die spezifische Information spezieller Figuren wird in davon abgeleiteten Klassen implementiert. Programmieren Sie die Klasse Kreis, Rechteck und Quadrat. Achten Sie auf guten Zugriffsschutz, d.h. alle Elemente müssen private sein, Zugriff erfolgt ausschließlich über Methoden. Konstruktoren sind vorhanden.
- b) Realisieren Sie in Ihrer main()-Funktion Polymorphismus, indem Sie ein Array mit verschiedenen Figuren anlegen und dann in einer Laufschleife Ihre Figuren mittels print() ausgeben.

Aufgabe 4

(SockenSortieren.java)

Sortieren Sie eine Schublade mit ca. 10 Socken der Größe nach. Socken sollen eine Farbe (String) und eine Größe (int) besitzen. Außerdem haben Socken das Interface Sortierbar implementiert, das folgendermaßen aussieht:

```
interface Sortierbar  
{  
    public int istGroesser(Sortierbar element);  
}
```

Die Methode ist Groesser() gibt eine negative Integer-Zahl, null oder eine positive Integer-Zahl zurück, je nachdem, ob das aktuelle Objekt kleiner, gleich oder größer als das angegebene Objekt ist.

Als Sortieralgorithmus verwenden Sie Bubblesort. Schreiben Sie hierzu Ihr Programm aus einer früheren Übungsaufgabe allgemein für Objekte vom Typ "Sortierbar" um.