



Übung 12

Thema: Vererbung

Ein Reeder möchte seine Flotte von Schiffen mit einem Java-Programm verwalten.

Teilaufgabe 1

Ein Schiff enthält die Information über den Schiffsnamen, den Kapitän und die Anzahl der Besatzung. Dabei wird übrigens der Kapitän mitgezählt. Der Kapitän ist ein Objekt der Klasse Person. Diese Klasse speichert den Vornamen, Nachnamen und das Geburtsdatum einer Person. Alle drei Werte müssen bei der Erzeugung eines Objekts vorliegen und sind nicht veränderbar.

Legen Sie für jede Instanzvariable eine Property an. Alle Variablen müssen Werte erhalten. Schreiben Sie einen Konstruktor zum Initialisieren der Instanzvariablen mithilfe der Properties.

Außerdem speichert jedes Schiff eine List bestehend aus Touren. Zu einer Tour gehören die Informationen zu dem Starthafen, Zielhafen, Abfahrtsdatum und Ankunftsdatum. Alle Daten müssen gültige Werte enthalten. Im Anhang befindet sich eine Liste der Häfen. Zu einem Schiff können beliebig viele Touren abgespeichert werden.

Die Klasse Schiff soll vorschreiben, dass alle von ihr abgeleiteten Klassen eine Methode mit Namen TourFestlegen haben.

Teilaufgabe 2

Es dürfen keine Schiffobjekte erstellt werden können! Es gibt nur Frachtschiffe und Ausflugsboote.

Frachtschiffe enthalten zu dem Schiffsdaten die zusätzlichen Informationen zur maximalen Ladekapazität (eine ganze Zahl, z.B. eine Angabe in Tonnen) und zur aktuellen Lademenge. Legen Sie dafür Properties mit Plausibilitätsprüfungen an.

Schreiben Sie einen Konstruktor zum Initialisieren aller Instanzvariablen beim Anlegen eines neuen Objekts. Schreiben Sie eine Methode Entladen, die eine zu entladende Menge als Parameter übergeben bekommt. Falls die zu entladende Menge kleiner oder gleich der aktuellen Lademenge ist, wird die aktuelle Lademenge entsprechend verringert. Ist die Menge größer als die aktuelle Lademenge, so wird die restliche Ladung entladen und die aktuelle Lademenge auf null gesetzt; mit einer entsprechende Hinweismeldung am Bildschirm, dass eben nicht genug vorhanden war.

Schreiben Sie eine Methode Beladen, die eine zu ladende Menge als Parameter übergeben bekommt. Zunächst wird geprüft, ob das Schiff leer ist, sonst wird gar nichts geladen. Falls das Schiff leer ist, wird das Schiff mit so viel Tonnen beladen, wie es der Parameter Menge vorschreibt, höchstens allerdings mit der maximalen Ladekapazität.

Denken Sie an die Methode Tourfestlegen.

Teilaufgabe 3

Erstellen Sie die Klasse Ausflugsboot. Außer dem Kapitän befinden sich maximal 20 Passagiere auf einem Ausflugsboot. Von jedem Passagier ist der Name und das Geburtsdatum bekannt. Schreiben Sie einen Konstruktor zum Initialisieren aller Instanzvariablen beim Anlegen eines neuen Objekts.

Schreiben Sie eine Methode MittelalterBerechnen. Diese Methode berechnet das Durchschnittsalter der aktuellen Passagierliste und gibt es zurück.

Denken Sie an die Methode TourFestlegen. Touren mit maximal 5 Personen dauern immer einen Tag. Sonst kann die Tour länger dauern. Die Passagierliste bezieht sich immer auf die letzte Tour.

Teilaufgabe 4

Erstellen Sie eine Factoryklasse Flotte. Die Klasse besitzt eine Dictionary bestehend aus Schiffsobjekten. Schlüssel der Collection ist der Schiffsname.

Lesen Sie aus der Datei alle Daten in das Programm.

Sorgen Sie zusätzlich dafür, dass alle Daten aus der Dictionary wieder in die Datei geschrieben werden.

Teilaufgabe 5

Erstellen Sie das Klassendiagramm.

Schreiben Sie ein Testprogramm:

Berechnen Sie für alle Ausflugsboote das Durchschnittsalter der letzten Tour.

Beladen Sie alle Frachtschiffe mit 300 Tonnen.
Entladen Sie 400 Tonnen von allen Frachtschiffen
Entladen Sie nochmal 400 Tonnen.

Zeigen Sie für alle Schiffe die Tourlisten an.

Legen Sie für ein Frachtschiff und für ein Ausflugsboot jeweils eine weitere Tour an.

Verlassen Sie die Anwendung und rufen Sie die Anwendung erneut auf.

Zeigen Sie für alle Schiffe die Tourlisten an. Die beiden neuen Touren müssen noch vorhanden sein.
