

Programmieren 1

Übungsserie 3

Stoff

- Bis und mit Kapitel 4
 - Fokus: Eigene Klassen Programmieren
-

Allgemeine Informationen zur Abgabe

- Die Abgabe erfolgt online auf ILIAS.
 - Quellcode zu den Implementationsaufgaben muss als ***.zip** Datei abgegeben werden. Exportieren Sie hierzu Ihr Projekt direkt aus Eclipse. Quellcode, den wir nicht kompilieren können, wird nicht akzeptiert.
 - Lösungen zu den Theorieaufgaben muss als ***.pdf** Datei abgegeben werden. Andere Formate werden nicht akzeptiert.
 - Arbeit in Zweiergruppen: Geben Sie jeweils nur ein Exemplar der Lösung pro Gruppe ab. Geben Sie in jeder Quellcode-Datei die *Namen und Matrikelnummern* beider Gruppenmitglieder in den ersten beiden Zeilen als Kommentar an.
 - Vorbesprechung: *13.10.2023*
 - Abgabe: *27.10.2023 13:00*
-

Implementationsaufgaben

1. Programmieren Sie eine Klasse **Thermometer**, welche einen einfachen Fieberthermometer modelliert. Die Klasse soll eine Temperatur in Celsius als einzige Instanzvariable speichern. Der Konstruktor soll diese Instanzvariable standardmässig auf *37.0* Grad setzen. Schreiben Sie eine Methode **increase**, welche die Temperatur um *0.1* Grad erhöht und einen *Getter* für die Temperatur. Zudem definieren Sie eine Methode **reset**, welche die Temperatur wieder auf *37.0* zurücksetzt.

Schreiben Sie eine zweite Klasse **ThermometerTest**, in der Sie zwei Objekte vom Typ **Thermometer** instanzieren und deren Methoden ausführlich testen.

2. Programmieren Sie eine Klasse **Car**, welche die Marke, das Modell und den Jahrgang des Fahrzeuges modelliert. Der Konstruktor soll diese drei Instanzvariablen gemäss Parameterübergabe initialisieren – zudem schreiben Sie *Getter* und *Setter* für alle Instanzvariablen und eine **toString** Methode für eine einzeilige Repräsentation von **Car** Objekten. Schliesslich definieren Sie eine Methode **isAntique**, welche einen **boolean** zurückgibt, der anzeigt ob das Auto aktuell älter ist als 45 Jahre (Durchsuchen Sie die API Dokumentation nach Möglichkeiten, das aktuelle Jahr vom System zu erfragen). In einer zweiten Klasse **Garage** instanziiieren Sie drei **Car** Objekte und testen alle programmierten Methoden.

3. Schreiben Sie eine Klasse **Cargo**, welche ein Stückgut mit Länge, Breite, Höhe und einem Namen modelliert (z.B. 30, 44, 65, "Kaffeemaschine"). Schreiben Sie einen Konstruktor, *Getter* und *Setter* für alle Instanzvariablen und eine Methode **toString**.

Schreiben Sie eine Klasse **Box**, die Instanzvariablen für die Länge, Breite und Höhe einer Box enthält. Zusätzlich enthält die Klasse **Box** eine Instanzvariable **full** vom Typ **boolean**, die angibt, ob die Box gefüllt ist oder nicht, sowie eine Instanzvariable **cargo** vom Typ **Cargo**. Der Konstruktor setzt die Länge, Breite und Höhe einer Box gemäss Parametern – neu instanziierte **Box** Objekte sollen standardmässig leer sein. Definieren Sie einen zweiten Konstruktor ohne Parameter, der eine Standard-Box mit Länge, Breite und Höhe 1 generiert. Zusätzlich definieren Sie eine Methode **getCapacity**, die das Volumen der Box berechnet und zurückgibt. Schliesslich schreiben Sie eine Methode **addCargo**, welche ein Objekt vom Typ **Cargo** als Parameter entgegennimmt. Falls dieses Stückgut gemäss Länge, Breite und Höhe in die Box passt, passen Sie die Variable **full** und die Instanzvariable **cargo** an und geben **true** zurück (andernfalls **false**).

Testen Sie die Klasse **Box**, indem Sie in einer weiteren Klasse **BoxTest** drei **Box** Objekte instanziiieren, manipulieren und ausgeben (hierzu benötigen Sie noch eine geeignete Methode **toString** in der Klasse **Box**, welche Angaben zu allen gespeicherten Daten zurückgibt).

4. Auf ILIAS (Übungen → Serie 3) finden Sie eine Datei **Book.java**. Ihre Aufgabe ist es die darin implementierte Klasse **Book** wie folgt zu erweitern:
 - (a) Schreiben Sie einen Konstruktor sowie *Getter* und *Setter* für alle Instanzvariablen.
 - (b) Implementieren Sie die Methode **age**, welche das Alter eines Buches (Anzahl Tage seit Erscheinungsdatum) berechnet und zurückgibt.
 - (c) Implementieren Sie die Methode **toString**, die alle Informationen eines **Book**-Objekts als String zurückgibt. Beispiel:
123, Die Blechtrommel, Günter Grass, 1.1.1959
 - (d) Vervollständigen Sie die Methode **input**, welche die Werte für **id**, **title**, **author** und **dateOfPublication** von der Kommandozeile vom Benutzer einliest und im jeweiligen **Book**-Objekt abspeichert. Ungültige Eingaben müssen Sie nicht abfangen.

Testen Sie sämtliche Methoden der Klasse **Book** in einer zusätzlichen Klasse **BookShelf**.

Hinweise:

- Zur Konvertierung zwischen **Date**- und **String**-Objekten verwenden Sie bitte die in **Book** vorhandenen Methoden **dateToString** und **stringToDate**.

- Objekte der Klasse `java.util.Date` bestehen im Wesentlichen aus einem `long`-Wert, der die Anzahl Millisekunden seit 00:00:00 GMT am 1.1.1970 angibt. Verwenden Sie diesen Wert in Ihrer Implementation der Methode `age`. Siehe auch Java API Dokumentation.
 - Beachten Sie, dass die Anzahl Millisekunden eines Jahres den Wertebereich von `int` übersteigt.
-