

Programmieren in Java

Einige **Übungen** zu den Kapiteln **Klassen und Objekte** und **Vererbung**

Aufgabe 1 (5+3+5+2+4+7+5+2+1 Punkte)

Ein Objekt der Klasse `Auto` führt Buch über den aktuellen Kilometerstand und den vorhandenen Kraftstoffvorrat.

Die Klasse hat die privaten Attribute

- `kmStand` (aktueller Kilometerstand, ganzzahlig),
- `verbrauch` (Verbrauch auf 100 Kilometern, reell),
- `tankVolumen` (Volumen des Kraftstofftanks, reell) und
- `kraftstoffVorrat` (aktueller Tankinhalt, reell).

Die Attribute werden von einem allgemeinen Konstruktor initialisiert.

Die Klasse besitzt die folgenden öffentlichen Operationen:

- Die Operation `info()` gibt den aktuellen Kilometerstand und den Tankinhalt aus.
- Die Operation `toString()` wandelt ein `Auto`-Objekt in eine Zeichenkette, die alle Attributwerte enthält, und liefert diese Zeichenkette als Ergebnis zurück.
- Die Operation `fahren()` liefert kein Funktionsergebnis. Sie „fährt“ die als Argument `km` vom Typ `int` übergebene Zahl von Kilometern. Der Kilometerstand wird angepasst und der Kraftstoffvorrat verringert sich entsprechend. Wir können natürlich nur so weit fahren, wie es der Kraftstoffvorrat im Tank zulässt.
- Die Operation `tanken()` liefert kein Funktionsergebnis. Das Argument `liter` vom Typ `double` gibt an, wie viel Kraftstoff nachgefüllt werden soll; der Tankinhalt nimmt entsprechend zu. Achtung: das Volumen des Tanks ist begrenzt; wir dürfen nur so viel Treibstoff auffüllen, dass der Tank nicht überläuft.

- a) Geben Sie eine UML-Spezifikation der Klasse `Auto` an, die auch die Argumente und Rückgabewerte der Operationen umfasst.
- b) Deklarieren Sie die Attribute der Klasse `Auto`.
- c) Implementieren Sie den allgemeinen Konstruktor der Klasse `Auto`.
- d) Implementieren Sie die Methode `info()`, die den aktuellen Kilometerstand und den Tankinhalt ausgibt.
- e) Implementieren Sie die Methode `toString()`, die ein `Auto`-Objekt in eine Zeichenkette wandelt, die alle Attributwerte enthält, und diese Zeichenkette als Ergebnis zurückliefert.
- f) Implementieren Sie die Methode `fahren()`. Sie „fährt“ die als Argument `km` vom Typ `int` übergebene Zahl von Kilometern. Der Kilometerstand wird angepasst und der Kraftstoffvorrat verringert sich entsprechend. Wir können natürlich nur so weit fahren, wie es der Kraftstoffvorrat im Tank zulässt.
- g) Implementieren Sie die Methode `tanken()`. Das Argument `liter` vom Typ `double` gibt an, wie viel Kraftstoff nachgefüllt werden soll. Achtung: das Volumen des Tanks ist begrenzt; wir dürfen nur so viel Treibstoff auffüllen, dass der Tank nicht überläuft.
- h) Erzeugen Sie ein Objekt der Klasse `Auto` namens `goggoMobil`; der Kilometerstand soll 0 km sein, der Verbrauch auf 100 Kilometern 5.0 Liter, das Volumen des Tanks 50 Liter und der aktuelle Kraftstoffvorrat 30 Liter.
- i) „Fahren“ Sie mit dem `goggoMobil` 300 Kilometer. Tanken Sie anschließend 45 Liter Kraftstoff.

Aufgabe 2 Vererbung (8 Punkte)

Die Klasse `Omnibus` erweitert die Klasse `Auto` um das private Attribut `anzahlSitze`, das die Anzahl der Sitzplätze eines Omnibusses angibt. Die Klasse besitzt einen allgemeinen Konstruktor, der die Attributwerte initialisiert. Die Operation `toString()` wandelt ein `Omnibus`-Objekt in eine Zeichenkette, die alle Attributwerte enthält, und liefert diese Zeichenkette als Ergebnis zurück.

Implementieren Sie die Klasse `Omnibus`. Überlegen Sie, welche Operationen der Basisklasse `Auto` Sie dabei verwenden können.