

Objektorientierte Programmierung

Grundlagen

Übung 7

Klassen, Objekte, Vererbung

Aufgabe 1

Ein Objekt der Klasse `Flugzeug` führt Buch über den Stand des Flugstundenzählers und den vorhandenen Kerosinvorrat.

Die Klasse hat die folgenden privaten Attribute, die alle vom Typ `double` sind:

- `maxGewicht` (maximales Startgewicht des Flugzeugs (kg)),
- `reiseGeschw` (Reisegeschwindigkeit (km/h)),
- `flugStunden` (Flugstunden laut Flugstundenzähler (h)),
- `verbrauch` (Kerosinverbrauch pro Flugstunde (l/h)),
- `tankKapazitaet` (Gesamtkapazität der Tanks (l)) und
- `kerosinVorrat` (in den Tanks vorhandenes Kerosin (l)).

Die Klasse besitzt u. a. die folgenden öffentlichen Operationen:

- Der allgemeine Konstruktor initialisiert die Attribute; er stellt sicher, dass der Kerosinvorrat nicht größer als die Tankkapazität ist.
- Die Operation `info()` gibt den aktuellen Stand des Flugstundenzählers und den Tankinhalt aus.
- Die Operation `toString()` wandelt ein `Flugzeug`-Objekt in eine Zeichenkette, die alle Attributwerte enthält, und liefert diese Zeichenkette als Ergebnis zurück.
- Die Operation `reichweite()` liefert als Funktionsergebnis einen Wert vom Typ `double`, der angibt, wie viel Kilometer das Flugzeug mit dem aktuellen Kerosinvorrat fliegen kann.
- Die Operation `fliegen()` liefert einen Wert vom Typ `boolean` zurück. Das Argument `km` vom Typ `double` gibt an, wie weit geflogen werden soll. Falls die aktuelle Reichweite kleiner als das Argument `km` ist, wird nicht geflogen und der Wert `false` wird zurückgegeben. Andernfalls fliegen wir die angegebene Zahl von Kilometern; der Flugstundenzähler wird angepasst und der Kerosinvorrat verringert sich entsprechend; der Wert `true` wird zurückgegeben.
- 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.

Implementieren (auf deutsch: schreiben) Sie die Klasse `Flugzeug`.

Schreiben Sie eine Methode `main()`, die ein Objekt der Klasse `Flugzeug` namens `albatros` erzeugt; das maximale Startgewicht soll 68000 kg sein, die Reisegeschwindigkeit 800 km/h; der Flugstundenzähler wird mit 500 Stunden initialisiert; der Kerosinverbrauch beträgt 2000 Liter pro Flugstunde, das Volumen der Tanks 20000 Liter und der aktuelle Kerosinvorrat ist 16000 Liter.

Fliegen Sie mit dem `albatros` 3000 Kilometer weit. Tanken Sie anschließend 5000 Liter Kerosin.

Testen Sie Ihre Klasse `Flugzeug` auch anhand anderer Beispiele.

Aufgabe 2

Die Klasse `FrachtFlugzeug` erweitert die Klasse `Flugzeug` um das private Attribut `maxZuladung` vom Typ `double`, das die maximale Zuladung eines Frachtflugzeugs angibt.

Die Klasse besitzt einen allgemeinen Konstruktor, der die Attributwerte initialisiert. Der Konstruktor stellt sicher, dass die maximale Zuladung nicht größer als Startgewicht minus Tankkapazität ist. (Das Gewicht des eigentlichen Flugzeugs wird hier nicht berücksichtigt.)

Die Operation `toString()` wandelt ein `FrachtFlugzeug`-Objekt in eine Zeichenkette, die alle Attributwerte enthält, und liefert diese Zeichenkette als Ergebnis zurück.

Implementieren Sie die Klasse `FrachtFlugzeug`. Überlegen Sie, welche Operationen der Basisklasse `Flugzeug` Sie dabei verwenden können.

Testen Sie Ihre Klasse `Frachtflugzeug`.