

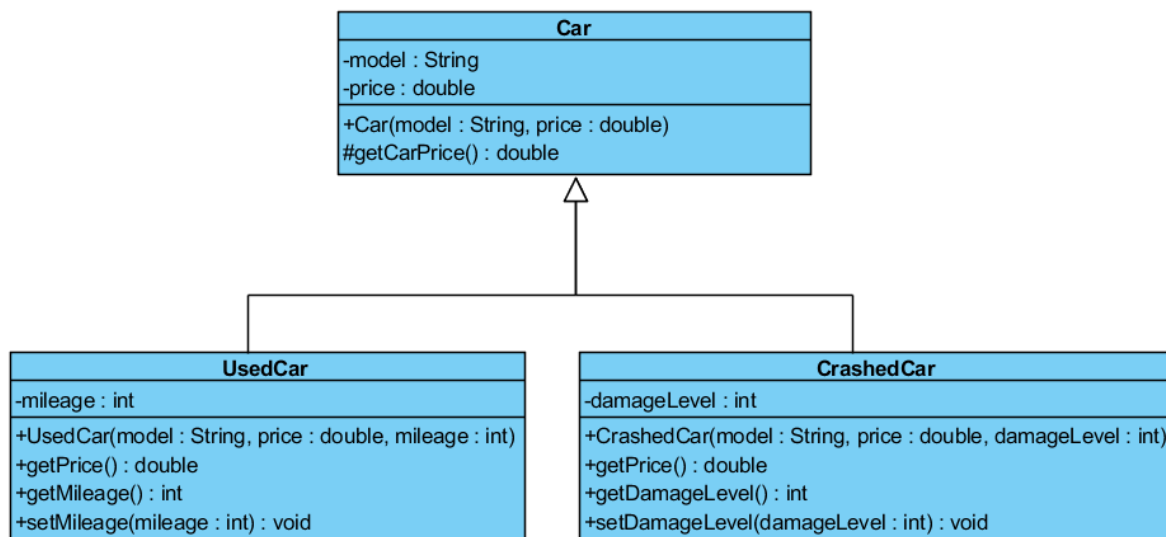
Übung Vererbung

Sie haben nun eine Einführung in die Vererbung erhalten. Es gilt nun, das gehörte in Form dieser Übung zu vertiefen.

Im nachfolgenden Beispiel geht es um Autos und die Berechnung des Werts bzw. Preises.

1. Alle Autos haben generell die Eigenschaften Modell und Preis.
2. Im Speziellen gibt es gebrauchte Autos und Unfallwagen.
3. Gebrauchte Autos weisen zusätzlich zu den generell gültigen Eigenschaften einen Km-Stand auf (mileage).
4. Unfallwagen haben zusätzlich ein Schweregrad des Schadens (damageLevel).
5. Der Wert von Gebrauchten Autos und Unfallwagen wird unterschiedlich berechnet:
 - Gebrauchtwagen: pro 10'000 km verkleinert sich der Wert um 5% des Neuwerts (Neuwert = price)
 - Unfallwagen:
 - Schweregrad 0 (low): Wert = Neuwert verkleinert sich um 10%
 - Schweregrad 1 (medium): Wert = Neuwert verkleinert sich um 50%
 - Schweregrad 2 (high): Wert = Neuwert verkleinert sich um 90%
 - Schweregrad 3 (total): Wert = 0

Es ist nun Ihre Aufgabe, diesen Sachverhalt zu programmieren. Als Hilfe erhalten Sie nachfolgend ein entsprechendes Klassendiagramm, welches Sie für Ihre Zwecke anpassen dürfen.



Beschreibung zum Klassendiagramm siehe nächste Seite.

Methodenbeschreibung

Klasse Car:

- `getCarPrice()`: liefert den Wert der Instanzvariable `price` zurück.

Klasse UsedCar:

- `getMileage()`: liefert den Wert der Instanzvariable `mileage` zurück.
- `setMileage(...)`: setzt einen neuen Km-Stand für dieses Auto
- `getPrice()`: berechnet den Wert des Autos aufgrund der Beschreibung in Punkt 5

Klasse CrashedCar:

- `getDamageLevel()`: liefert den Wert der Instanzvariable `damageLevel` zurück.
- `setDamageLevel(...)`: setzt einen neuen Schweregrad (Werte von 0-3)
- `getPrice()`: berechnet den Wert des Autos aufgrund der Beschreibung in Punkt 5

Testen

Testen Sie Ihren Code mit einer JUnit Testklasse mit Fokus darauf, ob der Wert des Autos in den unterschiedlichen Varianten (Gebrauchtwagen, Unfallwagen) korrekt berechnet wird.

- Test 1: `testPriceUsedCar`
 - o Initialisierung UsedCar Objekt mit Km-Stand 0
 - o Überprüfung ob Wert korrekt berechnet wird
 - o Veränderung Km-Stand auf entsprechende Grenzwerte
 - o Nach Veränderung jeweils Überprüfung auf Korrektheit des Werts mit `assert-` Methoden
- Test 2: `testPriceCrashedCar`
 - o Initialisierung CrashedCar Objekt mit `damageLevel` 0
 - o Überprüfung ob Wert korrekt berechnet wird
 - o Veränderung der `Damagelevel` auf alle möglichen Werte
 - o Überprüfung ob Wert jeweils korrekt berechnet wird