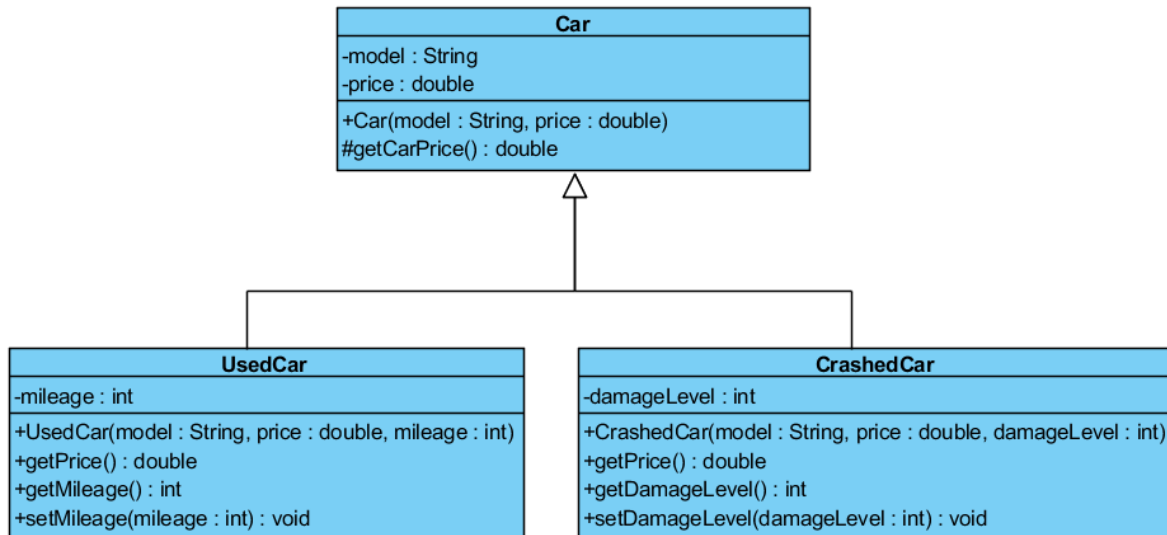


Arbeitsblatt Polymorphie

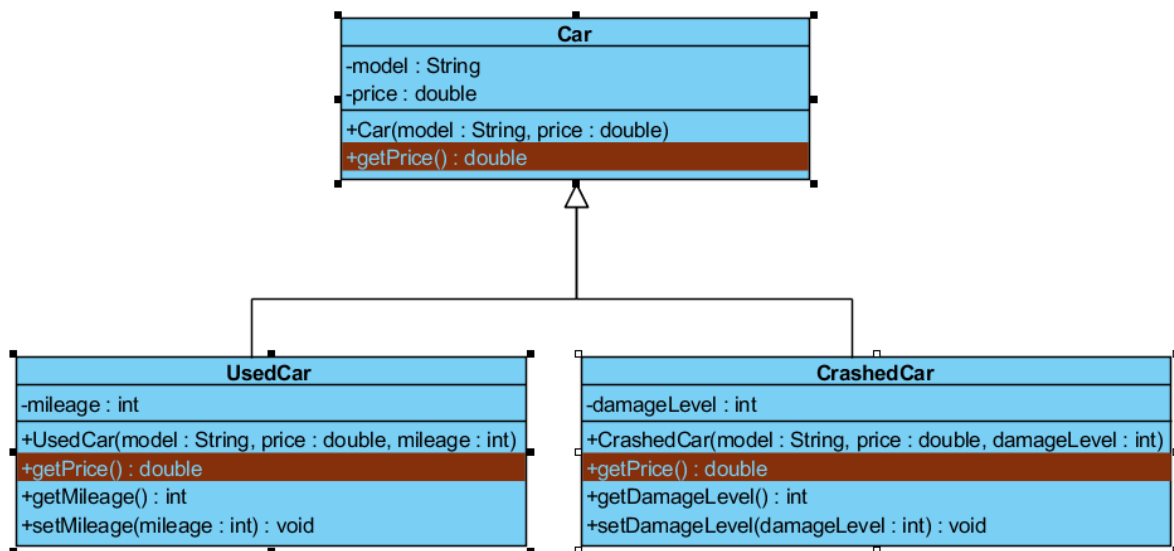
Ziele:

Mit Hilfe dieser praktischen Übung sollen Sie die Theorie noch weiter vertiefen.

Wir behalten das bestehende Beispiel "Car" bei. Im Folgenden nochmal das **Klassendiagramm der letzten Übung**:



Es gilt nun, dieses Beispiel mit **starker Polymorphie** zu lösen. Somit kommen Sie zu folgendem Klassendiagramm:



Um das Beispiel mit starker Polymorphie zu lösen, müssen Sie die Methoden `getCarPrice()` der Klasse `Car` bzw. `getPrice()` der Unterklassen entsprechend anpassen.

Wichtiger Hinweis:

Müssen Sie in einer Unterklasse explizit eine Methode der Oberklasse aufrufen, welche Sie in der Unterklasse aber überschrieben haben, können Sie dies mit `super.<Methodenname>(...)` realisieren.

Testen

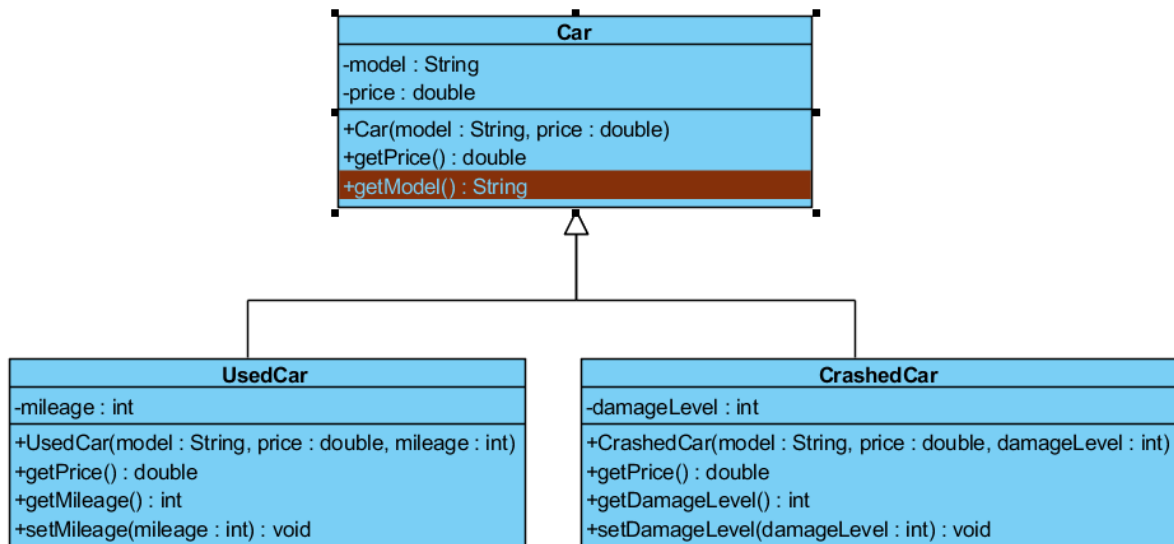
Testen Sie Ihren Code anhand einer Testklasse mit der main-Methode.

Bauen Sie Ihre Tests wie folgt auf:

- Test 1:
 - o Initialisierung UsedCar Objekt mit Km-Stand 10000
 - o Statischer Datentyp der Variable soll "Car" sein, dynamischer Datentyp UsedCar
 - o Überprüfung auf Korrektheit von getPrice()
- Test 2:
 - o Initialisierung CrashedCar Objekt mit damageLevel 0
 - o Statischer Datentyp der Variable soll „Car“ sein, dynamischer Datentyp CrashedCar
 - o Überprüfung auf Korrektheit von getPrice()

Erweiterung

Erweitern Sie die Klasse Car um einen getter für die Instanzvariable model.



Ergänzen Sie ihre Testklasse wie folgt:

- Test 3
 - o Initialisierung UsedCar Objekt mit Km-Stand 10000
 - o Initialisierung CrashedCar Objekt mit damageLevel 0
 - o Erstellung einer ArrayList, welche **Car** Objekte aufnehmen kann
 - o Die beiden Objekte der ArrayList hinzufügen
 - o ArrayList mit for oder foreach durchlaufen und für jedes Auto das Model, sowie den Preis auf die Konsole ausgeben.

Was beobachten Sie nach Ausführung dieses Tests und was schliessen Sie daraus?
