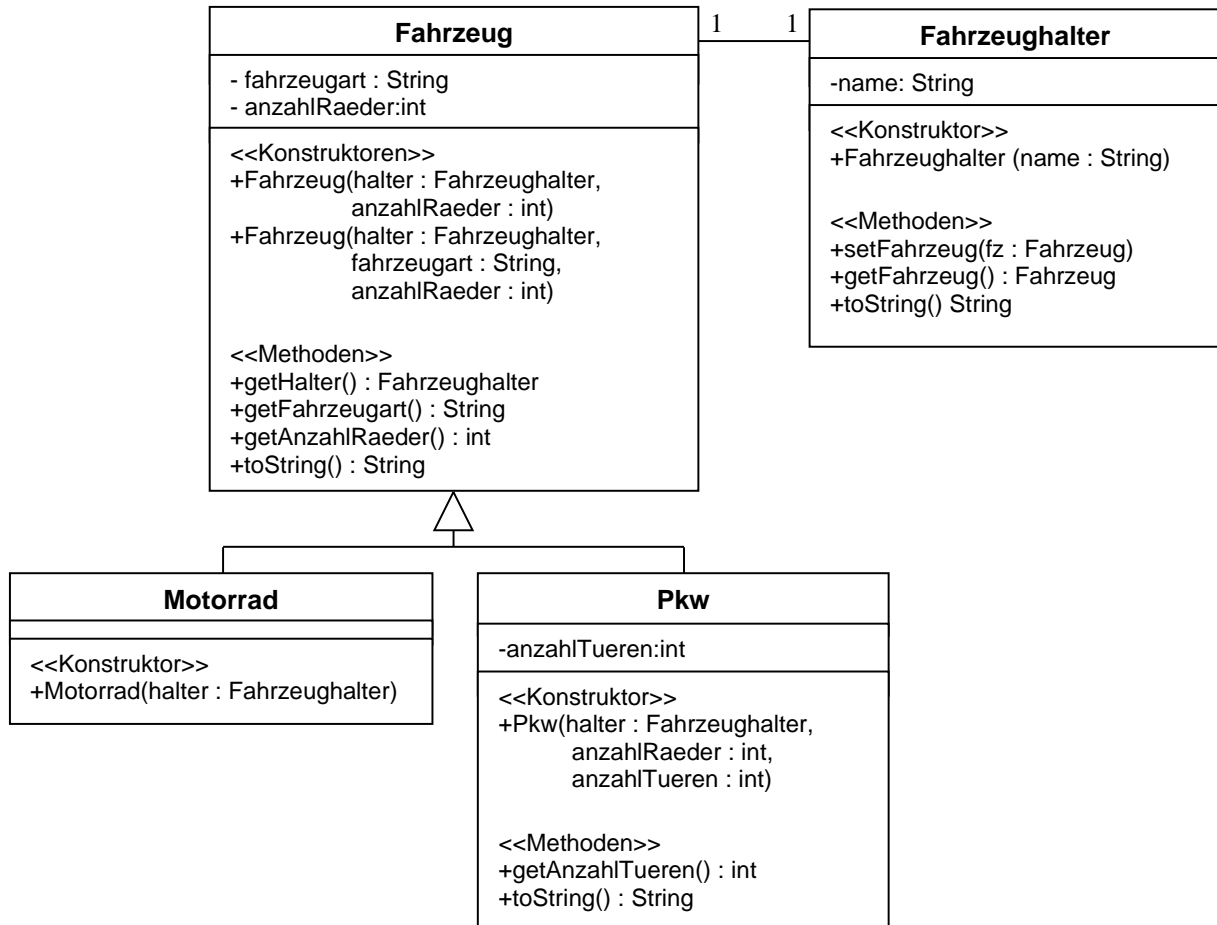


Übungen – Abstraktion und Beziehungen

Aufgabe 1 – Beziehungen

Gegeben ist das folgende erweiterte Klassen-Diagramm:



Den linken Teil, Fahrzeug, Pkw und Motorrad, kennen Sie bereits. Neu hinzugekommen ist die Klasse Fahrzeughalter und die Beziehung zwischen Fahrzeug und Fahrzeughalter.

Die bisherigen Klassen müssen entsprechend angepasst werden (Konstruktoren).

Es fällt auf, dass die Beziehung Fahrzeughalter–Fahrzeug zwar in UML modelliert wird, als Attribut **aber nicht** dargestellt ist. Sie müssen also Vorkehrungen treffen, dass diese gegenseitige Beziehung in Form von Attributen erstellt wird. Die Verbindung vom Fahrzeughalter zum Fahrzeug soll im Konstruktor der Klasse Fahrzeug hergestellt werden.

Beachten Sie, dass die meisten Klassen eine `toString()`-Methode haben. Achten Sie darauf, dass bei der Implementierung der `toString()`-Methoden, dort wo es angebracht ist, die `toString()`-Methode der Superklasse mit einbezogen wird.

Testen Sie in einer separaten Klasse die 1:1-Beziehung zwischen Fahrzeug und Fahrzeughalter. Testen Sie beide Richtungen der bidirektionalen Beziehung. Es könnte z.B. folgende Ausgabe resultieren:

```
Das Fahrzeug von Riese Goliath ist ein allgemeines Fahrzeug mit 16 Rädern.
Dieses Fahrzeug ist ein allgemeines Fahrzeug mit 16 Rädern. Es gehört Riese Goliath.

Das Fahrzeug von Dagobert Duck ist ein Pkw mit 4 Rädern und 5 Türen.
Dieses Fahrzeug ist ein Pkw mit 4 Rädern und 5 Türen. Es gehört Dagobert Duck.

Das Fahrzeug von Donald Duck ist ein Motorrad mit 2 Rädern.
Dieses Fahrzeug ist ein Motorrad mit 2 Rädern. Es gehört Donald Duck.
```

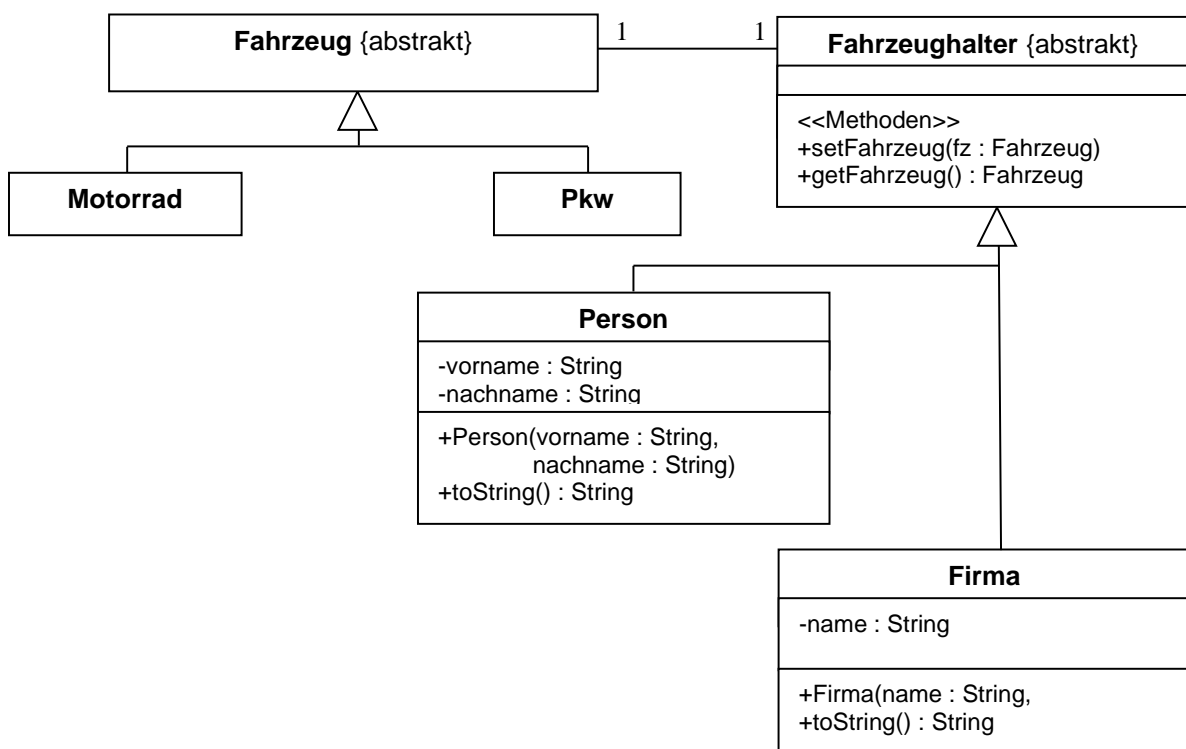
Aufgabe 2 – Abstrakte Klassen

Die Klasse Fahrzeug in unserem Vererbungsbeispiel ist lediglich ein Konzept, aber ein Objekt eines „allgemeinen Fahrzeuges“ anzulegen, eigentlich recht sinnlos. Machen Sie die Klasse Fahrzeug abstrakt und reagieren Sie auf Probleme, die nun mit dem bestehenden Testcode auftauchen.

Aufgabe 3 – Eine Hierarchie für Fahrzeughalter

Neben Personen können z. B. auch Firmen als Fahrzeughalter auftreten. Es bietet sich also an, die Klasse Fahrzeughalter als eine abstrakte Klasse zu deklarieren, um dann eine Klasse Person und eine Klasse Firma als Subklassen zu implementieren.

Wir haben also folgendes UML-Diagramm:



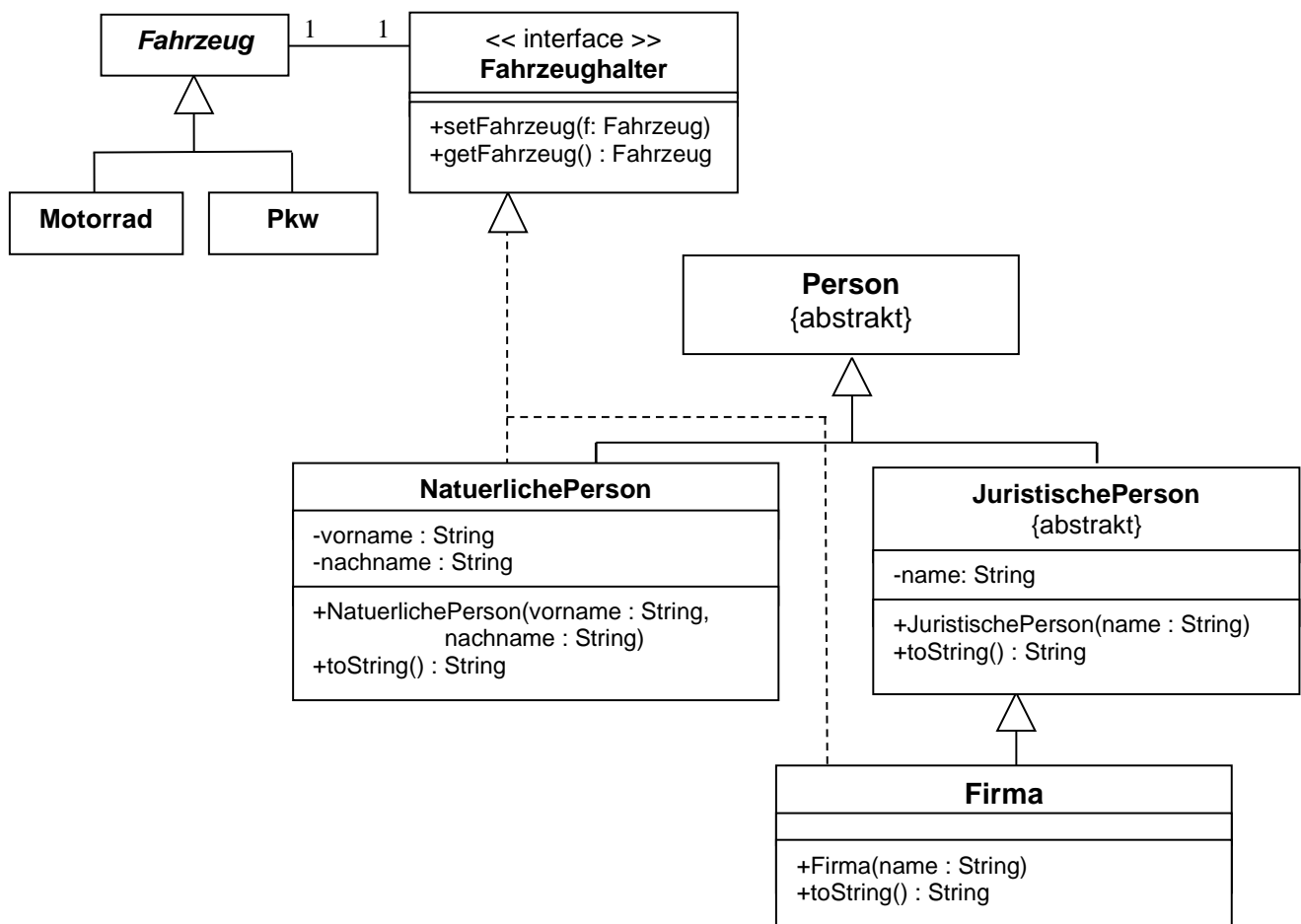
Erweitern Sie Ihr Testprogramm so, dass auch Firmen als Fahrzeughalter auftreten.

Das Fahrzeug von Firma Dagobert Inc. ist ein Pkw mit 4 Rädern und 5 Türen.
Dieses Fahrzeug ist ein Pkw mit 4 Rädern und 5 Türen. Es gehört Firma Dagobert Inc.

Das Fahrzeug von Donald Duck ist ein Motorrad mit 2 Rädern.
Dieses Fahrzeug ist ein Motorrad mit 2 Rädern. Es gehört Donald Duck.

Aufgabe 4 – Interfaces

Wir machen Fahrzeughalter zu einem Interface, so dass viele Klassen die Möglichkeit haben, ein Fahrzeughalter zu werden. Weiterhin ziehen wir eine zusätzliche Personen-Hierarchie ein, die bewusst nichts mit dem Interface zu tun hat (wir spielen hier aber nur...)



Beachten Sie, dass nicht alle Personen-Klassen das Interface implementieren. Juristische Personen sind im obigen Modell also nicht automatisch Fahrzeughalter, sondern nur eine Inkarnation davon, nämlich die Firmen.

Passen Sie den Testcode an, die Ausgabe sollte unverändert bleiben.