

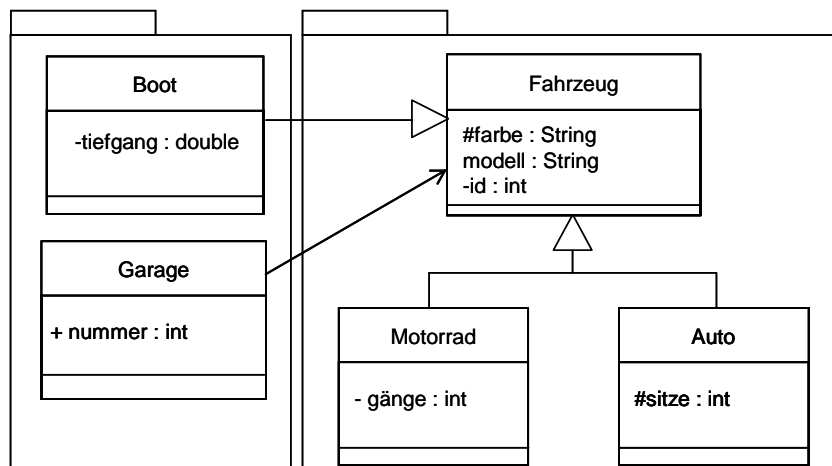


Übungsblatt 12

Aufgabe 28 *Sichtbarkeiten im Klassendiagramm*

(4 Punkte)

In der UML und in Java gibt es verschiedene Sichtbarkeiten. Geben Sie für folgendes Klassendiagramm an, welche unten stehenden Aussagen gelten.



- Autos können ihre Id auslesen.
- Motorräder können ihr Modell auslesen.
- Motorräder können ihre Farbe auslesen.
- Motorräder können ihre Gänge auslesen.
- Boote können ihre Farbe auslesen.
- Boote können ihr Modell auslesen.
- Garagen können die Farbe von Autos auslesen.
- Fahrzeuge können die Nummer ihrer Garage auslesen.

Aufgabe 29 *4+1 Sichtenmodell des RUP*

(6 Punkte)

- Beschreiben Sie zu jeder Sicht des 4+1 Sichten Modells von Philippe Kruchten, das dem Rational Unified Process (RUP) zugrunde liegt, jeweils mit einem Satz, welche Eigenschaften eines Software-Systems in der jeweiligen Sicht dargestellt werden.
- Nennen Sie jeweils eine Diagrammart, die sich für die Modellierung einer Sicht eignet.

Aufgabe 30 *Architekturmuster*

(4 Punkte)

In dieser Aufgabe soll die Software-Architektur für einen autonomen Roboter entworfen werden. Der Roboter ist dabei mit folgender Sensorik ausgestattet

- 2 Frontabstansensoren
- 1 Rückwärtsabstansensor
- 1 Lichtsensor zur Bestimmung der Helligkeit
- 1 Mikrophon

Der Roboter besitzt folgende Aktorik

- 2 Räder
- 1 Lampe

Zusätzlich ist er mit einem Funkmodul ausgestattet, das es erlaubt Funknachrichten zu senden und zu empfangen.

Für den Roboter ist bereits eine Basissoftware in Form eines kleinen Betriebssystems und ein TCP/IP-Netzwerkstack als Bibliothek zur Ansteuerung des Funknetzwerks vorhanden.

- a) Welches Architekturmuster eignet sich zur Strukturierung des Systems am besten? Begründen Sie Ihre Antwort kurz.

Die Vorverarbeitung der Sensordaten erfordert mehrere Schritte, die hintereinander ausgeführt werden müssen. Einzelne Schritte lassen sich für verschiedene Sensoren verwenden, andere wiederum sind spezifisch.

- b) Welches Architekturmuster eignet sich zur Strukturierung der Datenvorverarbeitung am besten? Begründen Sie Ihre Antwort kurz.

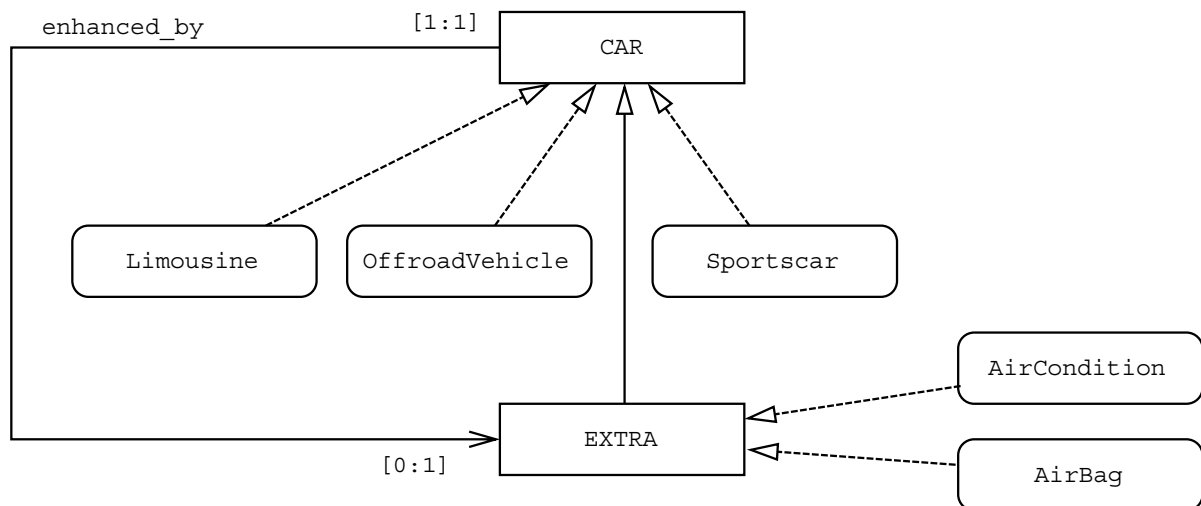
Aufgabe 31 *Graphspezifikation*

(6 Punkte)

Ausgehend von dem Beispiel aus Aufgabe 27c) soll in dieser Aufgabe eine PROGRES-Graphspezifikation für ein Werkzeug erstellt werden, mit dem Autos mit verschiedenen Extras modelliert werden können. Auf der Webseite zur Vorlesung sind Folien verfügbar, die noch einmal die wesentlichen Grundlagen für das *Spezifizieren mit Graphersetzungssystemen* beschreiben.

Die Spezifikation kann auf Papier erfolgen. Es kann jedoch auch der PROGRES-Editor verwendet werden. Auf der Webseite des Lehrstuhls ist eine Virtual Machine verfügbar, auf der die PROGRES-Umgebung installiert ist. Die zu vervollständigende Spezifikation kann von der Vorlesungswebseite heruntergeladen werden.

Für diese Aufgabe ist das folgende Graphschema vorgegeben.



Ergänzen Sie die PROGRES-Graphspezifikation um die folgenden Teile:

- die textuelle Spezifikation der Knotenklassen CAR und EXTRA mit dem intrinsischen Attribut `price` und dem abgeleiteten Attribut `Price`,
- die Transformation `CreateNewCar (carType : type_in CAR)`, die ein neues Auto eines gegebenen Knotentyps erzeugt,
- die Transformation `AddExtra(car : CAR, extraType : type_in EXTRA)`, die ein gegebenes Auto um ein Extra des angegebenen Typs erweitert.

Abgabe: 27.1.2009

Sie können Ihre Lösung zum obigen Termin bis spätestens 10:00 Uhr morgens am Lehrstuhl abgeben. Bitte vermerken Sie in jedem Fall die Namen und Matrikelnummern aller beteiligten Personen (maximal 3) und den Namen Ihrer Kleingruppenübung.

Aktuelle Informationen zur Vorlesung finden Sie auf den Webseiten des Lehrstuhls unter <http://se.rwth-aachen.de/swt>.