

---

## Software Engineering

<http://proglang.informatik.uni-freiburg.de/teaching/swt/2009/>

---

### Aufgabenblatt 8

#### Aufgabe 1: Collections in OCL (6 Punkte)

Sei *col* eine Collection in OCL. Implementieren Sie nun folgende Operationen.

1. **hasNElements**: Gibt **true** für eine Zahl *n* und einen Ausdruck *expr* zurück, falls es in *col* genau *n* Elemente gibt, die *expr* erfüllen. Dabei sei *it* die Iterationsvariable in *expr*.
2. **isUnique**: Gibt **true** zurück, falls *col* keine Duplikate enthält. Verwenden Sie *nicht* die eingebaute Funktion desselben Namens.
3. **take**: Gibt für eine Zahl *n* eine beliebige Teilmenge von *col* zurück. Die Größe der Teilmenge ist das Minimum aus *n* und der Größe von *col*.

#### Aufgabe 2: Vor-und Nachbedingungen in OCL (4 Punkte)

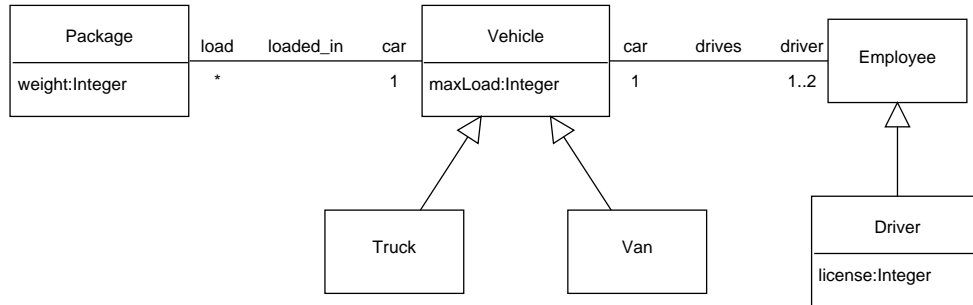
1. In der Vorlesung wurde eine Vorbedingung für die Operation **move** der Klasse **Meeting** angegeben. Verfeinern Sie diese Vorbedingung, so dass Meetings in verschiedenen Räumen zur selben Zeit stattfinden können.
2. Die Klasse **Meeting** aus der Vorlesung soll nun um eine Operation

`relocate(newLocation : Location)`

erweitert werden. Diese Operation ändert den Raum, in dem das Meeting stattfindet. Schreiben Sie sinnvolle Vor- und Nachbedingungen für **relocate**.

### Aufgabe 3: OCL in der Praxis (10 Punkte)

Das folgende Klassendiagramm modelliert einen Teil einer Speditionsfirma:



Implementieren Sie die folgenden Bedingungen in OCL:

1. Jeder Employee, der ein Vehicle fährt, muss von der Sorte Driver sein. (Es könnte noch andere Arten von Angestellten geben, die in dem Diagramm nicht dargestellt werden.)
2. Wenn das Vehicle ein Truck ist, dann werden ihm zwei Driver zugewiesen, Vans haben nur einen Driver.
3. Die Fahrer eines Trucks müssen eine License vom Wert 2 haben.
4. Das gesamte Gewicht aller Packages, die in einem Vehicle geladen sind, darf nicht größer sein als der `maxLoad` des Vehicles.