



## Formale Methoden

SS 2014 – Übungsblatt 9

Ausgabe: 18. Juni 2014

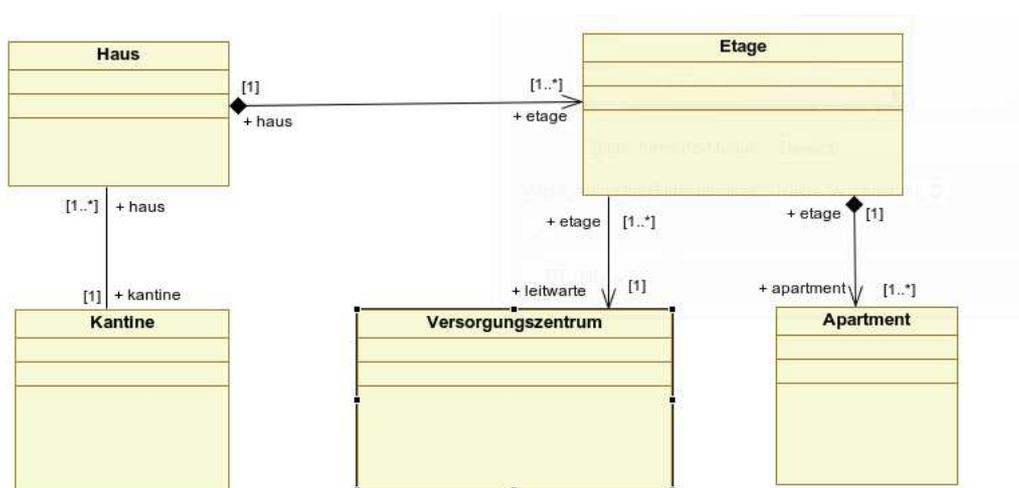
Abgabe bis 25. Juni 2014 an: [dsavvidi+fm@studs.math.uni-wuppertal.de](mailto:dsavvidi+fm@studs.math.uni-wuppertal.de)

### Aufgabe 1. Einfache OCL2-Datentypen/Ergebnistypen von Mehrfachnavigationen und collect()-Operationen

Lesen Sie den Abschnitt 7.4 der OCL2.4-Dokumentation. Erstellen Sie eine Übersicht über Werte/Literale und Operationen der *predefined* Typen.

Verifizieren Sie die Aussagen des Abschnitts 2.3 der Vorlesung über die Ergebnistypen von Mehrfachnavigationen und collect()-Operationen mit Hilfe von Papyrus am Beispiel des BankAccount-Modells. Beschreiben Sie dabei, wie Sie Papyrus zur Auskunft über die Typen von Navigationsergebnissen veranlassen können?

### Aufgabe 2. bedenklicher Contract



Konstruieren Sie in OCL2 die folgenden „Constraints“:

- Jedes (mit dem Default-Konstruktor `Haus()`) neu erzeugte Haus enthält noch keine Etagen und keine zuständige Kantine.

- Nach dem Hinzufügen einer Etage zu einem Haus mittels `Haus::addEtage(e : Etage)` enthält das Haus mindestens eine Etage.
- Nach dem Hinzufügen einer Etage zu einem Haus mittels `Haus::addEtage(e : Etage)` enthält das Haus eine Etage mehr als zuvor.

Welche Umkonzeptionen sind mindestens notwendig?

Wie könnte die Assoziation `apartment` der Klasse `Etage` effizienter modelliert werden?

### Aufgabe 3. *Student/Professor/Seminar/Belegung und eine Codevertragsverbesserung*

Spezifizieren Sie die Klassen und ihre Abhängigkeiten im UML-Modell `Student/Professor/Seminar/Belegung`:

<http://www.agilemodeling.com/artifacts/classDiagram.htm>

Ergänzen Sie je zwei sinnvolle virtuelle Attribute und Methoden (`OclHelper`).

Studieren Sie `Rental::newRental()` in Abschnitt 3.5 von [Object Constraint Language \(OCL\): A Definitive Guide](#).

Wie könnte der Codevertrag verbessert werden?

### Aufgabe 4. *OCL Tuple*

Lesen Sie Abschnitt 7.5.15 `Tuples` des OCL-Manuals. Welche Schwachstelle OCLs kann durch die Verwendung von `Tuples` ausgeglichen werden?

Erklären Sie in eigenen Worten den Unterschied zwischen abstrakter und konkreter Syntax einer Programmiersprache (vergleiche etwa

<http://www.cse.chalmers.se/edu/year/2011/course/TIN321/lectures/proglang-02.html>).

Wie sieht man diesen Unterschied in Kapitel 8 beziehungsweise 9 des OCL-Manuals wieder-  
gespiegelt?

### Aufgabe 5. *Umlpe*

Betrachten Sie das `Umlpe`-Beispiel

<http://cruise.eecs.uottawa.ca/umpleonline/?example=AirlineExample>.

Schildern Sie an diesem Beispiel, wie eine syntaktische Fassung von Assoziationen in neueren Programmiersprachen aussehen könnte und dann die Codeerzeugung im UML-Diagramm-Codegenerator viel einfacher implementiert werden könnte.