



**BERGISCHE
UNIVERSITÄT
WUPPERTAL**

Prof. Dr. Hans-Jürgen Buhl
Praktische Informatik/Numerik

Fakultät für
Mathematik und Naturwissenschaften,
Mathematik und Informatik

E-MAIL buhl@math.uni-wuppertal.de

WWW www.math.uni-wuppertal.de/~buhl

DATUM 4. Juli 2018

Formale Methoden

SS 2018 – Übungsblatt 11

Ausgabe: 5. Juli 2018

Abgabe bis 12. Juli 2018 an: <mailto:Daniel.Schiller@uni-wuppertal.de>

Aufgabe 1. *OCL-Constraints der Apartment-Wohnanlage*

Realisieren Sie die OCL-Constraints aus Abschnitt 2.18 der Vorlesung mit Hilfe von Papyrus. Benutzen Sie notfalls die schon diskutierten Workarounds! (Wo sind solche nötig?)

Aufgabe 2. *OCL-Constraints*

Konzipieren Sie einen Aufzählungstyp für Studierende, der Studierende als Gasthörer, als Seniorenstudierende beziehungsweise als Vollzeitstudierende ausweist (Zeichnung eines UML-Klassendiagramms).

Konzipieren Sie die folgenden OCL-Constraints:

- Wenn ein Studierender Gasthörer an einer Universität ist, muss er an einer anderen als Vollzeitstudierender immatrikuliert sein.
- Seniorenstudierende können (spezielle) Teilnahme-Scheine bekommen, alle anderen Studierenden müssen Leistungs-Scheine erwerben. (Konzipieren Sie dazu Methoden `getTeilnahmeschein()`, ... mit geeigneten Vorbedingungen.)
- Seniorenstudierende müssen mindestens 60 Jahre alt sein.
- Vollzeitstudierende müssen mindestens 12 Jahre alt sein.

Aufgabe 3. *Person::trageLebenspartnerschaftEin()*

Ergänzen Sie die Klasse `Person` um die Klasse `Meldeamt` mit einer Methode `trageLebenspartnerschaftEin()`. Überprüfen Sie mittels OCL-Constraints die Voraussetzungen für eine Lebenspartnerschaft: Nichtexistenz einer bestehenden, noch nicht aufgelösten Partnerschaft, Mindestalter, ... der Lebenspartner: Die Operation soll den Ergebniswert `true` liefern, falls die Lebenspartnerschaft eingetragen werden kann (dann auch Änderungen der Lebenspartnerschaftsklassenattributwerte, ..., in den Nachbedingungen spezifizieren).

Beachte, dass eine Person sich nicht als Lebenspartner von sich selbst eintragen können sollte.

Aufgabe 4. *Verwandschaftsbeziehungen als OclHelper*

Definieren und implementieren Sie die OclHelper-Methode

`istGrossOnkel (p: Model::Person): Boolean`

zur Abfrage, ob `p` Großonkel von `self` ist, sowie das OclHelper-Attribut

`Urgrossmutter: Set (Model::Person)`

für die Klasse `Person` mit Assoziationsenden/Rollennamen `elter` und `kind`.

Aufgabe 5. *M2-Abfragen von UML-Modellen*

Welche M2-Objekte existieren in UML-Modellen (etwa demjenigen von Aufgabe 1). Erstellen Sie eine Übersicht, indem Sie die automatische Vervollständigung des „interaktiven OCLs“ des UML-Editors benutzen.