



**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 7. Juli 2016

Formale Methoden

SS 2016 – Übungsblatt 11

Ausgabe: 7. Juli 2016

Abgabe bis 14. Juli 2016 an: 125319@uni-wuppertal.de

Aufgabe 1. *OCL-Constraints von Person (Personenstandsdaten)*

Realisieren Sie die OCL-Constraints aus Abschnitt 2.15f. 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 `get_Teilnahmeschein()`, ... mit geeigneten Vorbedingungen.)
- Seniorenstudierende müssen mindestens 60 Jahre alt sein.
- Vollzeitstudierende müssen mindestens 12 Jahre alt sein.

Aufgabe 3. *Person::trageEheEin()*

Ergänzen Sie die Fallstudie Personenstandsdaten um die Klasse Standesamt mit einer Methode `trageEheEin()`. Überprüfen Sie mittels OCL-Constraints die Voraussetzungen für die Eheerlaubnis: Werte von Status, Geschlecht, ... der Ehepartner: Die Operation soll den Ergebniswert `True` liefern, falls die Ehe eingetragen werden kann (dann auch Änderungen der Hochzeitsassoziationen, des Attributs Status der Ehepartner, ..., in den Nachbedingungen spezifizieren).

Aufgabe 4. *wochentag()*

Spezifizieren Sie die Nachbedingung der (neu eingeführten) Methode `wochentag()` der Klasse `Datum` mit Hilfe von `chronoJD(in d : Datum) : chronoJD`.

Aufgabe 5. *OclHelper*

Erklären Sie umgangssprachlich vier «OclHelper»-Beispiel-Konstrukte aus Abschnitt 7.4.4 des OCL 2.4-Manuals.

Schreiben Sie eine eigene OclHelper-Methode `next` mit zwei Parametern (einer `Sequence(T)`, einem gültigen Index in den ersten Parameter), der das auf den Index folgende T-Element als Rückgabewert liefert.

Wie könnte mit Hilfe dieser OclHelper-Methode das Constraint aus Abschnitt 2.14 der Materialsammlung kompakter formuliert werden?

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.