



**BERGISCHE
UNIVERSITÄT
WUPPERTAL**

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

Fachbereich C
Mathematik und Naturwissenschaften,
Mathematik und Informatik

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

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

DATUM 25. Juni 2014

Formale Methoden

SS 2014 – Übungsblatt 10

Ausgabe: 25. Juni 2014

Abgabe bis 2. Juli 2014 an: dsavvidi+fm@studs.math.uni-wuppertal.de

Aufgabe 1. *Person/Hypothek/Haus*

Bearbeiten Sie das Vorlesungsbeispiel Person/Hypothek/Haus mit Hilfe von Papyrus.

Verwenden Sie dabei den Standard-Workaround einer Papyrus-Assoziationsklasse durch eine (vorübergehende) zusätzliche `derived` Assoziation.

Aufgabe 2. *virtuelle Methoden für Datum*

Schreiben Sie geeignete virtuelle Methoden, mit denen Sie die Konstruktoren

```
+ $ Datum(cjd : chronoJD) : Datum  
+ $ chronoJD(d : Datum) : chronoJD
```

und die beiden Infix-Operatoren `minus` sowie `minusZinsTage` gemäß

http://de.wikipedia.org/wiki/Julianisches_Datum

sowie

<http://www.zinsen-berechnen.de/zinsmethoden/deutsche-zinsmethode.php>

nach der E30/360-Methode spezifizieren.

Aufgabe 3. *Redundanzen in der Klasse Euro*

Schreiben Sie OCL-Constraints, die die Infix-Operatoren `>=`, `<>`, `>` und `<=` auf `<` und `=` zurückführen.

Aufgabe 4. *count()/Guards*

Wie ist die Collection-Methode `count()` in OCL definiert?

Wie unterscheiden sich die Nachbedingungen für Sets von denen anderer Collection-Typen?

Beschreiben Sie in eigenen Worten die Benutzung von `count()` bei der Spezifikation der `union()` verschiedener Collection-Typen.

Ergänzen Sie die beiden OclHelper-Attribute `employer` und `employee` aus Abschnitt 2.4 der Materialsammlung jeweils um Guards für den Fall der Vielfachheit 0.

Aufgabe 5. *BankPersonCompany*

Verbessern Sie das BankPersonCompany-Model von Seite 101 der Materialsammlung/ Seite 23 des OCL 2.4 Manuals dadurch, dass Sie den Rollennamen `bankAccount` durch `bank` ersetzen, eine Assoziationsklasse `BankAccount` als Unterklasse einer abstrakten Klasse `Account` einfügen und für das private Attribut `Model::BankAccount::kontostand: Euro` geeignete Constraints formulieren.

Welche weiteren `properties` und/oder `operations` sollte `Account` enthalten?