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 18. Mai 2015

# Formale Methoden

**SS 2015 – Übungsblatt 5**
**Ausgabe: 19. Mai 2015**
**Abgabe bis 2. Juni 2015 an: *dsavvidi+fm@studs.math.uni-wuppertal.de***

**Aufgabe 1.** *Unterverträge*

Erläutern Sie, warum die Nachbedingung eines Modifikators einer „is-a“ Unterklasse im Falle der Gültigkeit der Vaterklassenvorbedingung nicht schwächer sein darf als die Vaterklassennachbedingung, jedoch andernfalls „beliebig“ sein darf:

```
------------------------------ Fussgaengerbruecke
   QUERIES
      MaxLast : REAL
      AktLast : REAL
   INVARIANTS
      MaxLast >= 7500
      AktLast <= MaxLast
   ACTIONS
      ueberquereBruecke( IN gew : REAL,
                         OUT Guthaben : INTEGER )
         PRE
            gew  + AktLast <= MaxLast
            gew <= 200
            Guthaben >= 2
         POST
            AktLast = OLD(AktLast) + gew
            Guthaben = OLD(Guthaben) - 2
      verlasseBruecke( IN gew : REAL )

         ...
```

sowie ein Subcontract:

```
------------------------------ Autobruecke
   QUERIES
      MaxLast : REAL
      AktLast : REAL
   INVARIANTS
      MaxLast >= 800000
      AktLast <= MaxLast
   ACTIONS
      ueberquereBruecke( IN gew : REAL,
                         OUT Guthaben : INTEGER )
         PRE
            gew  + AktLast <= MaxLast
            gew <= 20000
            Guthaben >= 20
         POST
            AktLast = OLD(AktLast) + gew
            OLD(gew) <= 200     IMPLIES Guthaben = OLD(Guthaben) - 2
            NOT OLD(gew) <= 200 IMPLIES Guthaben = OLD(Guthaben) - 20
      verlasseBruecke( IN gew : REAL )
         ...
```

**Aufgabe 2.** *Ein UML-Modell*

Konzipieren und konstruieren Sie ein Klassenmodell im Umfeld Bestellung, Lieferschein und Rechnung.

**Aufgabe 3.** *Softwarefehler*

Welche Fehler führten in http://www.heise.de/newsticker/meldung/44621 zu einer Katastrophe? Welche konstruktiven Maßnahmen hätten dieser vorbeugen können?

**Aufgabe 4.** *isEmpty(), DataType*

Wie wird in OCL (2.3.1) die abgeleitete Abfrage `isEmpty()` für die Collection `Set` in Form einer Nachbedingung spezifiziert, wie die Funktion `floor()` für `Real`?
Suchen Sie vier andere interessante Nachbedingungen und erläutern Sie diese.

Lesen Sie Kapitel 10.2 (DataTypes) in

UML2.5-beta.

Wie wird DataType erklärt? Welche Bedeutung hat der kleine gefüllte Punkt am Rollenende `DataType::ownedAttribute` beziehungsweise am Rollenende `Property::datatype` in Figure 10.1?

Skizzieren Sie die Diagramme zweier Objekt-Instanzen der Klasse Person aus Figure 10.3 oben genannten Dokuments, die beim Vergleich den Wert `true` liefern.

**Aufgabe 5.** *Heartbleed und welche Tools es (nicht) verhindern konnten*

Lesen Sie

How to Prevent the next Heartbleed

und erstellen Sie eine Übersicht über die dort bespochenen Software-Qualitätssicherungstools (jeweils Angabe, für was sie gut nutzbar sind).

Warum ist der Einsatz von Formalen Methoden auf dem Proof-Level nach

Heartbleed and Formal Methods

nicht sinnvoll. Was sollte stattdessen genutzt werden, SQA zu realisieren?