BERGISCHE UNIVERSITÄT
WUPPERTAL
GAUSS-STRASSE 20
42119 WUPPERTAL
TELEFAX (0202) 439-2901
TELEFON (0202) 439-0
WWW www.uni-wuppertal.de

Fachbereich C

MATHEMATIK UND NATURWISSENSCHAFTEN
Fachgruppe Mathematik und Informatik

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

E-Mail: Hans-Juergen.Buhl@math.uni-wuppertal.de
WWW: http://www.math.uni-wuppertal.de/∼buhl

Formale Methoden

SS 2006 – optionales Übungsblatt (klausurähnlich)

Ausgabe: 12. Juli 2006

Aufgabe 1. *Invarianten*

Demonstrieren Sie an je einem kleinen Beipiel die VDM++-Syntax von Invarianten an Typen beziehungsweise an `instance variables`.

Zu welchen Zeitpunkten des Laufs eines Programms ist die Gültigkeit von Invarianten gewährleistet?

Wie ist die Beziehung von Klasseninvarianten in Elternklassen in Bezug auf Kindklassen?

Aufgabe 2. *(umgangssprachliche) Bedeutung von OCL-Ausdrücken*

Erläutern Sie in eigenen Worten (umgangssprachlich) die Bedeutung der folgenden OCL-Constraints bzw. Ausdrücke

- ```
  context Apartment
  Etage.Apartment[self.Raumnummer] = self
  ```

- ```
  context Haus
  inv: Etage.Haus->asSet() = Set { self }
  ```

- ```
  context Etage
  inv: Haus.Etage->includes(self)
  ```

- ```
  context Versorgungszentrum
  inv: Etage.Apartment[22]->size() = 1
  ```

- ```
  context Kantine
  ... Haus.Etage.Apartment
  ```
  ```

Aufgabe 3. *OCL-Constraints*

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

Konzipieren Sie die folgenden OCL-Constraints:

- Wenn ein Student Gasthörer an einer Universität ist, muss er an einer anderen als Vollstudent 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.
- Vollzeitstudenten müssen mindestens 12 Jahre alt sein.

Aufgabe 4. *(umgangssprachliche) Bedeutung von VDM++-Ausdrücken*

Erläutern Sie in eigenen Worten (umgangssprachlich) die Bedeutung der folgenden VDM++-Spezifikationen:

- ```
  class AccountDB
    types

      public
      digit = BankAccount`digit

    instance variables
      db : map (seq of digit) to CashAccount

    operations
      public
      RequestWithdrawal : (seq of digit) * nat ==> bool
      RequestWithdrawal (accnum,amt) ==
        return (db(accnum).RequestWithdrawal(amt))
      pre accnum in set dom db;
  end AccountDB
  ```

- ```
  class Directory
      instance variables
      my_keys : seq of KeyType := [];
      inv len my_keys = count;
      inv forall i, j in set inds my_keys &    -- keys are unique
                    i <> j => my_keys(i) <> my_keys(j);
      my_values : seq of ValueType := [];
      inv len my_values = count;

      has : KeyType ==> bool
      has(k) ==
  ```

```
        return (k in set elems my_keys);
```

–
```
    put : KeyType * ValueType ==> ()
    put(k, v) ==
        (
            my_keys := my_keys ^ [k];
            my_values  :=  my_values ^ [v];
            count := count + 1;
        )
    pre not has(k)
    post has(k) and
         value_for(k) = v and
         elems my_keys~  union {k} = elems my_keys and
         forall ki in set inds my_keys~ &
                 my_values~(ki) = value_for(my_keys~(ki));
```

Aufgabe 5. *VDM-Spezifikationen*

Konstruieren Sie VDM++-Spezifikationen für Klassen im Anwendungsfall einer Speicherverwaltung:

`Adressen` sind nichtnegative, durch Vier teilbare ganze Zahlen.

`Speicherbloecke` haben eine `Startadresse` und eine `Laenge`. Die `Laenge` ist ein Vielfaches von Vier. Speicherblöcke haben darüber hinaus ein redundantes Attribut `Endadresse`. (Welche Invariante erfüllen Endadresssen? Welchem Typ/welcher Klasse sollten sie angehören?)

Die `Speicherverwaltung` besitzt eine Liste von freien Blöcken sowie eine Liste von belegten Blöcken. Alle Blöcke sind „paarweise disjunkt“.