



## Programming by Contract

SS 2007 – Übungsblatt 6

Ausgabe: 24. Mai 2007

Abgabe: bis spätestens 6. Juni 2007  
in der Vorlesung

### Aufgabe 1. *Subcontracting*

Geben Sie in eigenen Worten an, wie sich die Vorbedingungen, die Nachbedingungen und die Invarianten in einer Kindklasse relativ zu denjenigen einer Elternklasse verhalten sollten, wenn eine "is-a"-Vererbung benutzt wird.

Geben Sie je ein Beispiel im Falle „Kunde/Stammkunde“.

### Aufgabe 2. *enum als Klasse*

Übersetzen Sie das Beispielprogramm [enum-Day5.cc](http://enum-Day5.cc) der Vorlesung und führen Sie ausreichend viele Tests durch.

Ändern Sie den Input-Operator so ab, dass er auch die Zeichenkette "Sonnabend" akzeptiert und testen Sie erneut.

Welche Software-Gütekriterien sind durch diese Änderung verbessert worden?

Erweitern Sie die Klasse um Methoden `istWochenendtag()`, Addition einer ganzen Zahl zu einem Day-Exemplar, Subtraktion zweier Day-Exemplare und testen Sie.

Welche Bedeutung könnte der genannten Addition bzw. Subtraktion zugrunde liegen? Spezifizieren Sie diese ausreichend genau.

Welche Vorteile hat die Klasse `Day` gegenüber einer reinen `enum`-Lösung mit eigenen Operatoren?

### Aufgabe 3. *newmat10*

Besorgen Sie sich die Bibliothek `newmat10` aus dem Internet:

<http://www.robertnz.net/download>

Lesen Sie die Installationsanweisung, installieren Sie die Bibliothek und bringen Sie dann mit Hilfe dieser Bibliothek ein Beispielprogramm zum Laufen.

Lesen Sie die Dokumentation <http://www.robertnz.net/nm10.htm#refer>.

Welche Software-Gütekriterien kann man bei Benutzung dieser Bibliothek besonderes gut erfüllen?

Schreiben Sie in paar Test-Hauptprogramme, übersetzen Sie diese und bringen Sie sie ebenfalls zum Ablauf.

#### Aufgabe 4. *Ackermann-Funktion*

Die Ackermann-Funktion wird definiert als

```
ackermann(n,m) =  
  If n = 0  
  Then m + 1  
  Else  
    If m = 0  
    Then ackermann(n-1, 1)  
    Else ackermann(n-1, ackermann(n, m-1))
```

Erstellen Sie eine C++-Implementierung. (Berücksichtigen Sie dabei den möglichen Integer-Overflow.)

Berechnen Sie die Ackermann-Funktion für:

$$(n, m) \in \{(1, 1), (2, 2), (3, 3)\}$$

Das Abfangen von Overflows ist bei der Ackermann-Funktion besonders wichtig, da sie sehr schnell wächst:  $ackermann(4, 2)$  besitzt bereits über 21000 Ziffern und  $ackermann(4, 4)$  ist größer als  $10^{10^{21000}}$ .

Schreiben Sie eine umgangssprachliche Spezifikation nach dem Muster auf Seite 40 der Materialsammlung.

#### Aufgabe 5. *arithmetischer Mittelwert*

Es werde der arithmetische Mittelwert durch

$$Z := (X + Y)/2$$

beziehungsweise durch

$$Z := X + (Y - X)/2$$

berechnet.

Wie unterscheiden sich die Ergebnisse dieser beiden Algorithmen voneinander? (Wann liefert jeder der beiden Algorithmen einen Wert ungleich unendlich, wann den Wert NaN?)

Wann sollte deshalb der Algorithmus 1, wann der Algorithmus 2 benutzt werden?