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

Programming by Contract

WS 2005/2006 – Übungsblatt 8

Ausgabe: 15. Dezember 2005

Abgabe: bis spätestens 22. Dezember 2005
in der Vorlesung
oder per E-Mail an `c.markmann@uni-wuppertal.de`

Aufgabe 1. *Nana-Contracts*

Schreiben Sie für alle Methoden der Klasse `day` (Aufgabe 5 von Übungsblatt 6), auch die von Ihnen selbst konzipierten `istWochentag()`, `operator+` und `operator-` (Nana-)Spezifikationen in der Form des Abschnitts 6.1 der Vorlesung.

Aufgabe 2. *Schleifeninvarianten*

a) Schreiben Sie eine Funktion x^f zur Berechnung (echt) gebrochener Potenzen von x (also: $0 \leq f < 1$) unter Benutzung der Funktionen `log()` und `exp()`.

b) Schreiben Sie einen entsprechenden Algorithmus ohne die Benutzung von `log()`. Er soll statt dessen die Beziehungen

$$x^f = (\sqrt{x})^{2f}$$
$$x^{1+f} = x \cdot x^f$$

nutzen.

Wie kann man eine Nachbedingung für den Algorithmus b) mittels des Ergebnisses von a) aufstellen?

Aufgabe 3. *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 $10^{10^{10^{21000}}}$.

Schreiben Sie auch hier eine Nana-Spezifikation nach dem Muster des Abschnitts 6.1 der Vorlesung.

Aufgabe 4. *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 sollte deshalb der Algorithmus 1, wann der Algorithmus 2 benutzt werden?

Aufgabe 5. *Raster*

Bei numerischen Approximationen wird häufig mit äquidistanten Gittern

$$\Delta x = \frac{b - a}{N}$$

gerechnet. Warum ist das effiziente Verfahren der Akkumulation

$$y = a + i * \Delta x \quad (i = 0, 1, ..., N)$$

dem Verfahren

$$y = \frac{a * (N - i) + i * b}{N}$$

unterlegen? Welche Werte von N sollte man bevorzugen? In welcher Reihenfolge sollte man in Analogie zur Mittelwertberechnung die einzelnen Rechenoperationen ausführen?