

## 2. Übung zur Vorlesung „Informatik I“

Abgabe: Montag, 10. November, 10:00 Uhr

Die Aufgaben sollen schriftlich im Schrein der Informatik (Hermann-Rodewald-Str. 3, Foyer) abgegeben werden. Für jede Übungsgruppe gibt es ein gesondertes Fach. Bitte geben Sie Ihren Namen, Ihre Matrikelnummer und Ihre Übungsgruppe auf Ihrer Abgabe an.

Zusätzlich sollen die Aufgaben über iLearn abgegeben werden:

<https://www-ps.informatik.uni-kiel.de/iLearn/>

Die Zugangsdaten zu iLearn erhalten Sie per E-Mail. Mails können entweder an Ihre private oder Ihre Institutsadresse gesendet werden. Zugang zu Ihrer Institutsadresse erhalten Sie per Webinterface unter folgender URL:

<https://www.informatik.uni-kiel.de/imap/>

Bitte schließen Sie sich vor der Abgabe der Aufgaben in iLearn mit Ihrem Übungsgruppenpartner zu einer Kleingruppe zusammen.

Bitte entschuldigen Sie, dass iLearn noch nicht stabil funktioniert. Gültig sind daher die Übungsaufgaben, die Sie in der Vorlesung bzw. auf der Webseite zur Übung erhalten. Wir hoffen, dass das Testen der Programmieraufgaben in iLearn in den nächsten Tagen funktionsfähig sein wird.

Die Programmierumgebung DrScheme kann auf den Institutsrechnern mit dem Kommando  
`/home/scheme/plt/bin/drscheme`  
gestartet werden.

**Die Globalübung findet ab dem 12.11.2008 statt!**

### Präsenzaufgabe 1

Die Funktionen `newif` und `fak` und `newfak` sind definiert durch

```
(define (fak x)
  (if (= x 0)
      1
      (* x (fak (- x 1)))))

(define (newif b x y)
  (cond (b x)
        (else y)))

(define (newfak x)
  (newif (= x 0)
         1
         (* x (newfak (- x 1)))))
```

Werten Sie die Terme `(fak 1)` und `(newfak 1)` schrittweise entsprechend dem Substitutionsmodell aus.

## Präsenzaufgabe 2

Die Funktion  $f : \mathbb{N} \mapsto \mathbb{N}$  sei definiert durch  $f(n) = \sum_{i=1}^n i^2$ .

- (a) Schreiben Sie eine Funktion (`f-rek n`), die  $f$  in einem rekursiven Prozess berechnet.
- (b) Schreiben Sie eine Funktion (`f-iter n`), die  $f$  in einem iterativen Prozess berechnet.
- (c) Geben Sie die Größenordnung für die Laufzeit und den Speicherverbrauch von (a) und (b) an.

## Aufgabe 3

8 Punkte

Die Funktionen `double` und `f` sind definiert durch

```
(define (double x) (+ x x))
```

```
(define (f x y) (if (= y 0) x y))
```

Werten Sie die folgenden Terme schrittweise entsprechend dem Substitutionsmodell aus:

- (a) `(double (+ (f (double 4) (- 6 (/ 48 8))) 13))`
- (b) `(- (f 0 (double (+ (double (- 13 5)) 7))) 4)`

## Aufgabe 4 (Programmieraufgabe)

8 Punkte

Ziel dieser Aufgabe ist es, die Kubikwurzel einer positiven reellen Zahl  $x$  mit Hilfe des Newton-Verfahrens zu approximieren. Das Newton-Verfahren definiert eine Folge  $(x_i)_{i \geq 0}$  mit

$$x_i \xrightarrow{i \rightarrow \infty} \sqrt[3]{x}$$

Die Folge ist definiert durch:

$$\begin{aligned} x_0 &= x \\ x_i &= \frac{x/x_{i-1}^2 + 2x_{i-1}}{3} \text{ falls } i > 0 \end{aligned}$$

Schreiben Sie eine Funktion (`kubikwurzel x e`), die die ersten  $k$  Folgenglieder berechnet, so dass  $|x_k^3 - x|$  kleiner als die Schranke  $e$  ist und dann  $x_k$  als Näherungswert für  $\sqrt[3]{x}$  liefert.

## Aufgabe 5 (Programmieraufgabe)

6 Punkte

Die folgende Anordnung von Zahlen ist als *Pascalsches Dreieck* bekannt:

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
  ...
```

Die Zahlen an den beiden Schenkeln des Dreiecks sind alle 1 und jede Zahl im Innern des Dreiecks ist die Summe der beiden schräglinks und schrägrechts darüberstehenden Zahlen.

Schreiben Sie eine Funktion (`pascal zeile pos`), die die Zahl, die in der Zeile `zeile` und der Position `pos` im *Pascalschen Dreieck* steht, in einem rekursiven Prozess berechnet. Dabei wird immer `pos ≤ zeile` vorausgesetzt. Die Spitze des Dreiecks hat die Koordinaten `zeile = 0` und `pos = 0`. Der Aufruf (`pascal 4 2`) soll also das Ergebnis 6 liefern.

Verwenden Sie für Ihre Lösung *nicht* die Fakultätsfunktion!