Christian-Albrechts-Universität zu Kiel
Institut für Informatik

Prof. Dr. M. Hanus, F. Reck

2. Übung zur Vorlesung „Programmierung“

Abgabe am Freitag, 06. November - 12:00

Von nun an 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/

Vor der Abgabe der Aufgaben müssen Sie sich im iLearn mit Ihrem Übungsgruppenpartner zu einer Kleingruppe zusammenschließen. Nachdem eine Abgabe getätigt ist, ist dies nicht mehr ohne weiteres möglich.

Falls Sie Ihre Lösungen zusätzlich schriftlich im Schrein der Informatik (Hermann-Rodewald-Str. 3, Foyer) abgeben, vergessen Sie bitte nicht Ihren Namen, Ihre Matrikelnummer und Ihre Übungsgruppe auf Ihre Abgabe zu schreiben.

Zum Starten der Programmierumgebung DrScheme auf den Institutsrechnern verwenden Sie folgendes Kommando:

`/home/scheme/plt/bin/drscheme`

Leider lassen sich nicht alle mathematischen Symbole, die in den Übungsblättern vorkommen, in den iLearn-Webseiten darstellen. Greifen Sie in diesen Fällen auf die PDF-Version zurück.

Präsenzaufgabe 1 - Fakultät

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 - Programmierübung

Implementieren Sie die folgenden beiden Funktionen:

(a) `(summe_1_bis n)`, die die Summe der ersten `n` natürlichen Zahlen berechnet. Gibt es außer der naiven Lösung auch eine weitere, effizientere?

(b) `(kgv n m)`, die das kleinste gemeinsame Vielfache zweier natürlicher Zahlen berechnet.

Aufgabe 3 - Kubikwurzel — 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

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

Die Folge ist definiert durch:

$x_0 = x$

$x_i = \frac{x/x_{i-1}^2 + 2x_{i-1}}{3}$ falls $i > 0$

Schreiben Sie eine Funktion `(kubikwurzel e x)`, 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 4 - Pascalsches Dreieck — 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` $\leq$ `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!

Aufgabe 5 - Auswerten — 6 Punkte

Die Funktionen `dec` und `g` sind definiert durch

```
(define (dec x) (- x 1))

(define (g x y) (if (= x y) x (g x (- y 2))))
```

Werten Sie die folgenden Terme schrittweise entsprechend dem Substitutionsmodell aus:

(a) `(g (dec (* 6 4)) (+ (+ 17 4) (dec 3)))`

(b) `(dec (g (dec (* 11 4)) (- (/ 100 2) 5)))`