ProgrammierParadigmen

Übung - Gruppe 6 & 7

Tobias Kahlert

Organisatorisches

- Tobias Kahlert
- E-Mail: tobias.kahlert@student.kit.edu
- Schreibt mich für Anregungen, Fragen und Feedback jederzeit jeder Zeit an
- Folien sind unter https://github.com/SrTobi/pp2017
- Wenn eure Fragen irgendwie auch für andere interessant sein könnten, benutzt das Forum im Ilias:
  - https://ilias.studium.kit.edu/ilias.php?ref_id=732265&...
  - + Ihr erreicht nicht nur einen Tutor, sondern Alle
  - + Wir haben Alle Benachrichtigungen eingerichtet
  - + Andere Studenten können auch von euren Fragen profitieren

Übungsblätter sind freiwillig, aber…

- Abgabe immer Donnerstag 11:30 Uhr
- Besprechung am nächsten Dienstag
- Abgaben auf Papier
  - Einwurf in die Briefkästen im UG von Geb. 50.34
- Abgabe von Quelltexten im Praktomat
  - *https://praktomat.cs.kit.edu/pp_2017_WS*
  - *Ist auch außerhalb des Uni Netzes verfügbar*

Nicht vergessen

- Nächste Woche Dienstag (31.10.2017) ist Feiertag
- Dafür <u>diese</u> Woche Freitag
  - Freitag, den 27.10.2016, 09:45-11:15 Uhr, Raum 236 Informatikgebäude (50.34)
  - Freitag, den 27.10.2016, 11:30-13:00 Uhr, Raum -119 Informatikgebäude (50.34)

… Klausur

- 120 Minuten
- 120 Punkte in 10 Aufgaben (1 Minute pro Punkt)
- Kofferklausur – Alles in Papierform darf mitgenommen werden
  - Vorlesungsfolien
  - Übungsblätter
  - Aufzeichnungen
  - Bücher
- Die Paradigmen der einzelnen Programmiersprachen zu verinnerlichen ist wichtig!!!
- In der Klausur gibt es keinen Compiler der auf euch Fehler hinweist!

(a) Definieren Sie eine Funktion [8 Punkte]

```
intermsl :: (a -> b -> a) -> a -> [b] -> [a]
```

so dass `intermsl` $\oplus$ i l für eine Liste $l = [l_0, l_1, l_2, \ldots]$ die Liste

$$r = [i,\ i \oplus l_0,\ (i \oplus l_0) \oplus l_1,\ ((i \oplus l_0) \oplus l_1) \oplus l_2,\ \ldots]$$

mit $r_0 =$ `i`, $r_{n+1} = r_n \oplus l_n$ berechnet. Dies soll auch für unendliche Listen funktionieren.

(b) Zur approximativen Lösung der durch f gegebenen Differentialgleichung [12 Punkte]

$$y'(x) = f(x, y(x))$$

mit dem Eulerverfahren bei Anfangswert $y(x_0) = y_0$ und Schrittweite h bildet man die Folgen $(y_n)_{n\in\mathbb{N}}$ und $(y'_n)_{n\in\mathbb{N}}$ der Funktionswerte bzw. Steigungen an den Stellen $x_n = x_0 + nh$, definiert durch:

$$\begin{aligned} y_{n+1} &= y_n + h y'_n \\ y'_n &= f(x_n, y_n) \end{aligned}$$

Hiermit hat y_n näherungsweise den Wert der gesuchten Funktion y an Stelle x_n: $y(x_n) \approx y_n$

Implementieren Sie das Eulerverfahren. Vervollständigen Sie hierzu die Funktion `euler`, indem Sie die Folgen $(x_n)_{n\in\mathbb{N}}$, $(y_n)_{n\in\mathbb{N}}$, $(y'_n)_{n\in\mathbb{N}}$ als unendliche Listen `xns, yns, yn's :: [Double]` definieren.

```
euler :: (Double -> Double -> Double) ->
          Double -> Double -> Double -> [Double]
euler f x0 y0 h = yns
  where xns  =
        yns  =
        yn's =
```

Hinweis: Unter anderem könnten `iterate`, `zipWith` und/oder `intermsl` nützlich sein.

Blatt 0 – Aufgabe 1

Installieren Sie die Haskel l Plattform auf Ihrem Rechner. Dies ist dringend empfohlen. Alternativ finden Sie unter

http://pp.ipd.kit.edu/lehre/misc/vm/virtualmachine.php?lang=de

die Programmierparadigmen-VM als Festplatten-Abbild. Diese enthält (neben der Haskell-Plattform) Vorinstallationen aller in der Vorlesung verwendeten Sprachen.

Blatt 0 – Aufgabe 2

Erstellen Sie eine Datei mit den Beispielprogrammen aus den Vorlesungsfolien und testen Sie diese, indem Sie sie im Interpreter *ghci* auf Beispielwerte anwenden.

Blatt 0 – Aufgabe 3

Definieren Sie eine Funktion

max3 x y z

welche das Maximum dreier Zahlen zurück gibt. Geben Sie dabei Varianten an, die ausschließlich

a) **if .. then .. else** verwendet

b) guard-Notation verwendet

c) die vordefinierte Funktion **max x y** verwendet

Listen und Endrekursion

a) Definieren Sie eine Funktion

    listSum xs

   welche über alle Elemente einer Liste iteriert und die Werte addiert.

b) Wandeln Sie diese in eine endrekursive Funktion

    listSum‘ xs

   um.

c) Testen Sie die beiden Funktion mit **Test.QuickCheck**

Haskell

https://xkcd.com/1312/