

Funktionale Programmierung

Übungsblatt (Abgabe: Mo., den 26.10. um 10:10 Uhr)

Prof. Dr. Margarita Esponda

Ziel: Auseinandersetzung mit der Haskell-Syntax, vordefinierten Haskell-Funktionen und ersten einfachen Funktionsdefinitionen.

1. Aufgabe (5 Punkte)

Verwenden Sie Haskell als Taschenrechner und berechnen Sie folgende Ausdrücke. Erläutern Sie kurz die Ergebnisse oder die Fehler, die dabei angezeigt werden.

div 1 2	8^200	2**1023	2**1024
8^0	4 /= 5	0.1 == 0.3/3	abs -7
rem 5 2	mod 5 2	rem 5 (-2)	mod 5 (-2)
sqrt 3	sqrt (-1)	exp 1	'z' < 'A'
"ba" < "aba"	True < False	True False	True && False

2. Aufgabe (8 Punkte)

Was ist der **Wert** und der **Datentyp** folgender Ausdrücke? Versuchen Sie, zuerst die Lösungen zu schreiben, ohne in dem Haskell-Interpreter die Ausdrücke einzugeben. Begründen Sie Ihre Antworten.

```
(-) ((+) ((+) 1 2) 3) (-2)
2**3 + (2^3)
log 1
div 1 2.0
mod 4 (-3)
(4 `mod` 5) == (4 `rem` 5)
(-4 `mod` 5) == (-4 `rem` 5)
(4 `mod` (-5)) == (4 `rem` (-5))
succ 4 * 8.0
succ (4 * 8)
if (mod 1 2) == 0 then "ja" else "nein"
True || undefined
True && not (True || undefined)
True && (undefined || True)
3/0 ** 2
2 - 0/0
```

3. Aufgabe (4 Punkte)

Schreiben Sie eine Haskell-Funktion, die bei Eingabe einer Anfangsinvestition, des Jahreszinssatzes und der Laufzeit in Jahren das Endkapital am Ende der Laufzeit berechnet. Verwenden Sie für die Berechnung folgende Formel, die die Berechnung der kontinuierlichen Verzinsung entspricht:

$$\text{Endkapital} = \text{Startkapital} \cdot e^{\text{Jahreszinssatz} \cdot \text{Laufzeit}}$$

Hier ist **e** gleich der Euler-Zahl, die mit Hilfe der **exp** Haskell-Funktion (**e = exp 1**) berechnet werden kann.

4. Aufgabe (4 Punkte)

Die Berechnung der Großkreis-Entfernung (*gke*) zwischen zwei Orten kann bei Eingabe der geographischen Breiten- und Längengrade (x_1, y_1, x_2, y_2) der Orte mit folgender Formel realisiert werden:

$$gke = c \cdot \arccos(\sin(x_1) \cdot \sin(x_2) + \cos(x_1) \cdot \cos(x_2) \cdot \cos(y_1 - y_2))$$

$$\text{mit } c = 111.2225685 = \text{Kilometerabstand zwischen zwei Längengraden}$$

Schreiben Sie eine Haskell-Funktion, die die Großkreis-Entfernung berechnet. Die Haskell Funktionen arbeiten mit Bogenmaß. Der Bogen- und Längengrad muss vorher umgerechnet werden. Definieren sie dafür die Funktionen **rad2grad** und **grad2rad**. Zum Testen berechnen Sie die Entfernung zwischen Berlin und Mexico City.

5. Aufgabe (3 Punkte)

Schreiben Sie eine Funktion, die ein Zeichen als Argument bekommt und entscheiden kann, ob das Zeichen ein Klammerzeichen ('(', ')', '[', ']', '{', '}') ist.

6. Aufgabe (8 Punkte)

In der Bildverarbeitung werden für die Darstellung von Farben unterschiedliche Formate verwendet.

Es gibt z.B. das RGB-Format, in dem mit Hilfe von drei ganze Zahlen zwischen 0 und 255 (Rot, Grün und Blau-Werte) die Farben kodiert werden.

Im Zeitschriften und Büchern wird das CMYK-Format verwendet, in dem Bilder aus einer Kombination der Farben Cyan, Magenta, Yellow und Black (CMYK) gedruckt werden.

Schreiben Sie eine Haskell-Funktion mit dem Namen **rgb2cmyk**, die unter Verwendung folgender Formel und nach Eingabe der RGB-Werte die entsprechenden CMYK-Werte berechnet.

Wenn $\text{RGB} = (0,0,0)$, dann ist $\text{CMYK} = (0,0,0,1)$,

sonst werden die Werte nach folgenden Formeln berechnet:

$$w = \max(R/255, G/255, B/255)$$

$$C = (w - (R/255)) / w$$

$$M = (w - (G/255)) / w$$

$$Y = (w - (B/255)) / w$$

$$K = 1 - w$$

Wichtige Hinweise:

- 1) Verwenden Sie geeignete Namen für Ihre Variablen und Funktionsnamen, die den semantischen Inhalt der Variablen oder die Semantik der Funktionen wiedergeben.
- 2) Verwenden Sie vorgegebene Funktionsnamen, falls diese angegeben werden.
- 3) Kommentieren Sie Ihre Programme.
- 4) Verwenden Sie geeignete lokale Funktionen und Hilfsfunktionen in Ihren Funktionsdefinitionen.
- 5) Schreiben Sie in alle Funktionen die entsprechende Signatur.