

Aufgabenblatt 1 – Haskell

Grundlage für die Bearbeitung folgender Aufgaben ist ein gründliches Studium und Durcharbeiten des bereitgestellten Skriptes von Hermann PUHLMANN: Funktionale Programmierung mit Haskell.

Aufgabe 0

Schreiben Sie für folgende „Probleme“ geeignete Funktionen und testen Sie diese. Alle Funktionen erwarten als Eingabe eine ganze Zahl.

- (a) *add5* addiert zu einer Zahl den Wert 5
- (b) *hoch3* berechnet die 3. Potenz einer ganzen Zahl
- (c) *inc* erhöht eine Zahl um eins.
- (d) *ist_zweistellig* liefert True, wenn die Zahl zweistellig ist

Aufgabe 1

Gesucht ist eine Funktion *bmi*, die (a) zu einem Gewicht (in kg) und Größe (in m) einer Person den BMI-Faktor bestimmt. Geben Sie die Typsignatur sowie die Funktionsdefinition an. Den BMI-Faktor errechnet man gemäß folgender Formel:

$$bmi = \frac{Gewicht}{Groesse^2}$$

(b) *bmi2*, zu einem Gewicht (in kg) und Größe (in **cm**). Ausgabe auf eine Nachkommastelle gerundet.

Aufgabe 2

Die Funktion *dreiMal* gibt einen als Parameter gegebenen String 3-mal hintereinander in einer Zeile auf dem Bildschirm aus. Geben Sie die Typsignatur sowie die Funktionsdefinition an.

Aufgabe 3

Eine Bakterienkultur bedeckt eine Fläche von 0,2 cm² und vermehrt sich jede Stunde um 5 %.

a) Wie hoch ist die bedeckte Fläche A(24) nach 24 Stunden?

Ansatz:

$$A(t) = A(0) \cdot \left(1 + \frac{p}{100}\right)^t$$

wobei A(0) = 0,2 cm², p Prozent und t Zeit in Stunden.

Definieren Sie eine Funktion *baktFlaeche*, die drei Parameter bekommt: die Ausgangsfläche in cm², die Vermehrungsrate in Prozent und die Zeit t in Stunden. Als Ergebnis wird die Fläche der Bakterienkultur ausgegeben.

Definieren Sie, falls notwendig, für die weiteren Aufgaben eine entsprechende Funktion oder benutzen Sie die definierte Funktion aus a) zur Beantwortung:

- b) Nach wieviel (vollen!) Stunden ist die Ausgangsfläche doppelt so groß?
- c) Bestimmen Sie die tägliche Zuwachsrate in %.
- d) Nach wie vielen Tagen wird eine Fläche von mindestens 80 cm² bedeckt sein?

Aufgabe 4

a) Die logischen Grundfunktionen *und* und *oder* sollen nachgebaut werden. Jede Funktion erhält als Parameter zwei boolesche Werte. Geben Sie die Typsignatur sowie die Funktionsdefinition an. Implementieren Sie beide Funktionen mit Hilfe des "Pattern Matching" Verfahrens (PUHLMANN, S. 14ff)

b) Implementieren Sie beide Funktionen (*und2*, *oder2*) mit Hilfe von „Guards“ (PUHLMANN, S. 16ff)

c) Implementieren Sie eine Funktionen *xor* nach dem Pattern Matching Verfahren. Die Funktion erhält zwei Parametern als **Char**: jeweils "w" für Wahr bzw. "f" für Falsch. Die Ausgabe ist wiederum ein Char: „w“ oder „f“

d) Implementieren Sie eine Funktion *nand* nach dem Pattern Matching Verfahren. Diese enthält als Parameter zwei Int: diese können den Wert 0 oder 1 annehmen. Die Ausgabe ist ein Int: 0 oder 1

e) Implementieren Sie nur unter Verwendung der *nand* Funktion aus d) die Funktionen *und3*, *oder3* und *xor3*.

f) Implementieren Sie einen eigenen komplexen logischen Ausdruck als Funktion mit zwei Parametern unter Verwendung der Definitionen aus e) und einer eigenen not3 Funktion.

Aufgabe 5

Bestimmen Sie ohne Computerhilfe den **Typ** und das **Ergebnis** folgender Ausdrücke. Begründen Sie ihr Ergebnis. Die verwendeten Funktionen entsprechen den Funktionen bei Aufgabe 0 und PUHLMANN, S. 7 und 8.

- a) (mod 8 6) + (div 8 6)
- b) (20 >= 4) && "See" == "SEE"
- c) ist_zweistellig (inc (add5 (length "123"))))
- d) mod (quadrat (inc 7)) 6