

Übungen zu Funktionaler Programmierung

Übungsblatt 2

Ausgabe: 20.10.2017, **Abgabe:** 27.10.2017 – 16:00 Uhr, **Block:** 1

Aufgabe 2.1 (4 Punkte) *Datentypen*

- a) Modellieren folgende Eigenschaften mit Datentypen. Geben Sie den Attributen Namen.
- Ein Konto hat einen Kontostand und einen Kunden als Besitzer.
 - Für einen Kunden werden die Daten Vorname, Name und Adresse (String) gespeichert.
- b) Legen Sie ein Beispielkonto als Tupel mit Konstruktor an (ohne Attribute).
- c) Legen Sie ein weiteres Beispielkonto als attribuiertes Tupel an.

Aufgabe 2.2 (2 Punkte) *Maybe*

Mit dem Datentyp `Maybe` können partielle Funktionen definiert werden. Definieren Sie eine Divisionsfunktion `div' :: Int -> Int -> Maybe Int`, welche bei der Division durch Null `Nothing` ausgibt.

Hinweis: Sie dürfen die Funktion `div` wiederverwenden.

Aufgabe 2.3 (3 Punkte) *Fallunterscheidungen nach Bedingungen*

Implementieren Sie folgende Funktionen in Haskell und geben Sie die Typen der Funktionen an.

a)
$$\text{collatz}(n) = \begin{cases} n/2, & \text{falls } n \text{ gerade} \\ 3n + 1, & \text{falls } n \text{ ungerade} \end{cases}$$

Sie können die Haskell-Funktion `div` und `even` benutzen.

b)
$$f(x, y) = \begin{cases} y * 2, & \text{falls } x = 0, y > 50 \\ y + 2, & \text{falls } x = 0, y \leq 50 \\ x * 2, & \text{falls } y = 0, x < 100 \\ x + y, & \text{sonst} \end{cases}$$

Aufgabe 2.4 (3 Punkte) *Präfixdarstellung*

Fügen Sie die impliziten Klammern in folgende Haskell-Ausdrücke ein. Wandeln Sie danach den Ausdruck in seine Präfixdarstellung.

a) $x + 3 * y * z$

b) $\text{add3 } 1 \ 2 \ 3$

c) $f \$ g . h \ x$