

(FP-Aufgaben 03)

Aufgabe 1: Welchen nicht-polymorphen Typ haben die folgenden Funktionen?

a) $f1\ x = x : [(1.3, 'a')]$

b) $f2\ x\ 0 = 0 : x$
 $f2\ x\ 1 = x$

Aufgabe 2: Warum ist das folgende Programm fehlerhaft?

$f\ 0\ x = []$
 $f\ x\ y = x$

Aufgabe 3: Schreiben Sie eine Funktion, die zwei Zahlenlisten als Parameter nimmt und die Zahlen, die an gleiche Positionen sind, zusammenaddiert. Die Ausgabe der Funktion muss eine Liste von Zahlen sein:

$addListen :: [Int] \rightarrow [Int] \rightarrow [Int]$

z.B. der Aufruf

$addListen\ [4, 3, 7]\ [5, 10]$

muss die Liste $[9, 13, 7]$ zurückliefern.

Aufgabe 4: Schreiben Sie ein Programm, das einen Geldbetrag in Cent als Parameter nimmt und berechnet mit welchen Münzen minimaler Anzahl der Geldbetrag bezahlt werden kann.

Zum Beispiel: 299 Cent ist

1 mal 200 Cent

1 mal 50 Cent

2 mal 20 Cent

1 mal 5 Cent

2 mal 2 Cent

Hinweis:

- Definieren Sie zwei Funktionen, damit die Rechnungen und Ausgaben getrennt bearbeitet werden:

$umrechnen :: Int \rightarrow [(Int, Int)]$

$ausgeben :: [(Int, Int)] \rightarrow [(Int, Int)]$

- Sie können die vordefinierten Funktionen **div** und **mod** verwenden.

Aufgabe 5: Schreiben Sie ein Programm, das die ersten n Fibonacci-Zahlen in eine Liste aufammelt (n ist eine ganze positive Zahl).

$fibListe :: Int \rightarrow [Int]$

Zum Beispiel: wäre n gleich 7, dann soll das Ergebnis $[0, 1, 1, 2, 3, 5, 8]$ sein.

Hinweis: Der **(++)**-Operator kann zwei Listen verketteten.

z.B. $[1, 2] ++ [32, 5, 9]$ liefert $[1, 2, 32, 5, 9]$ zurück.