**FP-Aufgaben 03**

Aufgabe 1

Welchen Typ haben die folgenden Haskell-Ausdrücke?

* **[(1.3,'a')]**
* **[ ]**
* **[(1.3, 'a')]** ist vom Typen **[(Double, Char)]**
* **[]** ist vom Typen generische Liste **[a]**

Aufgabe 2

Schreiben Sie eine Funktion, die das n-te bis m-te Element einer Liste ausgibt.

**unterListe :: Int → Int → [a] → [a]**

unterListe :: Int -> Int -> [a] -> [a]

unterListe n m ls =

take m (drop n ls)

Aufgabe 3

Schreiben Sie eine Funktion, die die ersten n Fibonacci-Zahlen in eine Liste aufsammelt.

**fibListe :: Int → [Int]**

fibListe :: Int -> [Int]

fibListe n

| n < 0 = []

| otherwise = fibListe (n - 1) ++ [fib n]

fib :: Int -> Int

fib 0 = 1

fib 1 = 1

fib x = fib (x -1) + fib (x -2)

Aufgabe 4

Schreiben Sie eine Funktion, die eine Liste von Paaren in ein Paar von Listen umwandelt.

**listeZuPaar :: [ ( String , Int ) ] -> ( [ String ] , [ Int ] )**

listeZuPaar :: [(String, Int)] -> ([String], [Int])

listeZuPaar [] = ([], [])

listeZuPaar (x : xs) =

case x of

(str, zahl) ->

(str : fst (listeZuPaar xs), zahl : snd (listeZuPaar xs))