technische universität dortmund

FLDIT

Prof. Dr. Peter Padawitz

Lehrstuhl 1
Logik in der Informatik

Jos Kusiek (*jos.kusiek@tu-dortmund.de*)
Lukasz Czajka (*lukasz.czajka@tu-dortmund.de*)

Wintersemester 2018/2019

Übungen zu Funktionaler Programmierung
Übungsblatt 5

Ausgabe: 9.11.2018, **Abgabe:** 16.11.2018 – 16:00 Uhr, **Block:** 2

Das Übungsblatt behandelt Themen bis einschließlich Folie 83.

Aufgabe 5.1 (12 Punkte) *Unendliche Listen auswerten*
Werten Sie folgende Haskell-Ausdrücke *schrittweise* und *lazy (leftmost-outermost)* aus.

a) `take 2 $ nats 3`

b) `iterate tail [3,4,9,8] !! 2`

c) `fibs !! 2`

Aufgabe 5.2 (6 Punkte) *Unendliche Listen implementieren*
Implementieren Sie folgende unendliche Listen. Die Listen dürfen keine Endlosschleifen mit `take` oder `(!!)` erzeugen.

a) `odds :: [Int]` – Liste aller ungeraden Zahlen.
Beispiel: `take 10 odds` ⇝ `[1,3,5,7,9,11,13,15,17,19]`

b) `alternate :: [Int]` – Liste aller Ganzzahlen ohne Null als alternierender Reihe. Die Liste soll mit 1 starten.
Beispiel: `take 10 alternate` ⇝ `[1,-1,2,-2,3,-3,4,-4,5,-5]`

c) `solutions :: [(Int, Int, Int)]` – Liste *aller* Tripel $(x, y, z) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$, welche die Gleichung $5x + 3y^2 + 10 == z$ lösen. Nehmen Sie für x, y und z nur positive Werte.
Beispiel: `(25,4,183) `elem` solutions` ⇝ `True`

Aufgabe 5.3 (6 Punkte) *Modellierung*
Gegeben seien folgende Datentypen:

```
type ID = Int
type Bank = [(ID,Account)]
data Account = Account { balance :: Int, owner :: Client }
  deriving Show
data Client = Client
  { name :: String
  , surname :: String
  , address :: String
  } deriving Show
```

Definieren Sie folgende Funktionen. Fehlerbehandlungen sind nicht notwendig.

a) `credit :: Int -> ID -> Bank -> Bank` – Addiert den angegebenen Betrag auf das angegebene Konto.

b) `debit :: Int -> ID -> Bank -> Bank` – Subtrahiert den angegebenen Betrag von dem angegebenen Konto.

c) `transfer :: Int -> ID -> ID -> Bank -> Bank` – Überweist den angegebenen Betrag vom ersten Konto auf das zweite.