6. Arbeitsblatt

Ausgabe: 28.01.99
Bearbeitung: bis Vorlesungsende

Shi Hui
Kolyang
Axel Lankenau
Christoph Lüth
Lutz Zegartowski

Literatur:

„The Craft of Functional Programming“, Kapitel über *Generalization*, *Functions as Values* und *Classes in Haskell*

Leitfragen:

1. Was sind Funktionen höherer Ordnung? Gibt es entsprechende Konzepte in anderen Programmiersprachen, die Sie kennen?

2. Was ist der Unterschied zwischen den folgenden Funktionstypen? Nennen Sie Beispiele!

```
(Int -> String) -> Int
Int -> (String -> Int)
Int -> String -> Int
(Int, String) -> Int
(a -> String) -> String
```

3. Diskutieren Sie Methoden der Vererbung, des Überschreibens und des Überladens von Funktionen.

Anwendungsaufgaben:

1. Betrachten Sie die auf Bäumen operierende Funktion:

```
mapTree :: (t -> u) -> Tree t -> Tree u
```

   Implementieren Sie diese und die zugehörigen Funktionen `sumTree` und `foldTree`. Orientieren Sie sich dabei an den Funktionen `map`, `foldr`, `foldl`.

2. Diskutieren Sie die Abbildung von Listenkomprehension auf `map`, `filter` und `fold` an Beispielen.

3. Definieren und implementieren Sie die Funktion

```
powerSet :: Set t -> Set (Set t)
```

   die die Menge aller Untermengen zurückliefert.

4. Implementieren Sie folgende Funktionen:

```
zip :: [t] -> [u] -> [(t,u)]
unzip :: [(t,u)] -> ([t],[u])
zipWith :: (t -> u -> v) -> [t]-> [u]-> [v]
```