

Höhere Funktionen

1 Das Problem

Ein wichtiger Aspekt der Softwareentwicklung ist die Bearbeitung von Daten. Und dies soll möglichst geschickt und effizient geschehen. Es geht in diesem Abschnitt um ein wichtiges Konzept der funktionalen Sprachen, was heute schon viele Sprachen übernommen haben. Man übergibt Funktionen Funktionen als Parameter.

2 Einige Beispiele

```
1 malzwei [] = []
2 malzwei (x:xs) = (2*x):malzwei xs

4 weristgerade [] = []
5 weristgerade (x:xs) = odd x:weristgerade xs

7 addierewas [] = []
8 addierewas (x:xs) = (x+3) : addierewas xs

10 nurgroessernull [] = []
11 nurgroessernull (x:xs) | (x>0) = x:nurgroessernull xs
12                          | otherwise = nurgroessernull xs
```

3 Die map-Funktion

Man sieht, dass in jedem Beispiel auf jedes Element der Liste eine Operation ausgeführt wird und so die gesamte Liste mit dieser Funktion bearbeitet wird. Dafür gibt es die map-Funktion.

Listing 1: Signatur und Implementierung der map-Funktion

```
1 map (a -> b) -> [a] -> [b]
2 map f [] = []
3 map f (x:xs) = f x : map f xs
```

Man sieht an der Signatur (a -> b), dass eine Funktion als Parameter übergeben wird. Es können auch mehrstellige Funktionen (a -> b -> c) verwendet werden.

4 Die elegante Implementierung der Beispiele

```
1 malzwei l = map (2*) l

3 weristgerade l = map (odd) l

5 -- Implementiere die anderen beiden Funktion mit map
```

Hinweis: Man kann sogar noch die Liste l weglassen. Es gilt die Regel, was rechts und links in einer Implementierung gleich ist kann weggelassen werden. Der Rechner sieht ja an der Signatur, dass eine Liste als Argument hinzukommt.

Man kann auch zwei Funktionen kombinieren.

```
1 mapListe = map ((*2) . quad )
2           where quad x = x*x
```

5 Übungen

```
2 -- Aufgabe: Studiere die folgende Beispiele:
3 f1 = map abs [-1,-3,4,-12]
4 f2 = map snd [("Eins",1),("Zwei",2),("Drei",3),("Vier",4),("Fuenf",5)]
5 f3 = map (Data.Char.digitToInt) "987654"
7 -- Aufgaben: Implementiere eine Funktion f4, die in einem String alle
  Buchstaben gross macht.
8 -- Implementiere eine Funktion f5, die aus einem Text alle Vokale
  herausstreicht.
```

6 Noch eine höhere Funktion

Häufig sucht man auch in Listen nach bestimmten Dingen. Z.B. finde alle Elemente größer Null. Dafür gibt es die Funktion `filter`.

Listing 2: Definition von `filter`

```
2 filter :: (a -> Bool) -> [a] -> [a]
3 filter bedingung [] = []
4 filter bedingung (x:xe) | (bedingung x) = x:(filter bedingung xe)
5                          | otherwise = filter bedingung xe
```

7 Beispiele und Aufgaben

```
2 fil1 = filter (>5) [1,2,3,4,5,6,7,8]
3 fil2 = filter odd [3,6,7,9,12,14]
4 fil3 = filter (\x -> length x > 4) ["aaaa","bbbbbbbbbbbbbb","cc"]
6 -- Aufgabe: istklein, nurpositive
```