



29. Dezember 2017

Haskellskripte

Dargestellt sind einige Lösungen der Aufgaben im Skript *Haskelleinführung*.

Listing 1: Rucksackproblem

```
1  -- Loesungen zu den Uebungen vom Blatt Haskelleinfuehrung.pdf
2
3  -- Eine Testliste zum Testen
4  liste = [1,3,5,2,12,33,8]
5
6  -- Aufgabe 1: Reste bei Division durch eine Zahl
7
8  -- todo
9
10 -- Aufgabe 2: teilerliste
11
12 teilerliste :: Int -> [Int]
13 teilerliste n = [ i | i <- [1..n], mod n i == 0 ]
14
15 -- Aufgabe 3: primliste
16
17 -- todo
18
19 -- Aufgabe 4: pythliste: Liste aller pythagoraeischen Tripel
20
21 pyTriple n = [(x,y,z) | x <- [2..n], y <- [x+1..n], z <- [y+1..n], x*x+y*y
    ==z*z]
22
23 -- Aufgabe 5: Anzahl eines bestimmten Elementes in einer Liste
24
25 -- Aufgabe 6: Das Collatz-Problem
26
27 collatz 1 = 1
28 collatz n = if even n then collatz (n `div` 2) else collatz (3*n+1)
29
30 collatz2 1 = [1]
```



```
31 collatz2 n = if even n then (n `div` 2):collatz2 (n `div` 2) else
    (3*n+1):collatz2 (3*n+1)
32
33 -- Collatzfolge mit List Comprehensions
34 -- collatzlist n = [:collatz (n `div` 2) | even n]
35
36 -- Aufgabe: Verwendung von foldr und lambda-Funktion
37
38 foo = foldr (\x y -> (x+y)/2) 54 [12,4,10,6]
39
40 -- Aufgabe: Arbeitsweise von list comprehensions
41
42 pairs :: [a] -> [b] -> [(a,b)]
43 pairs xs ys = [ (x,y) | x<-xs , y<-ys ]
44
45 -- Aufgabe: quad und mal 42
46
47 mapListe = map ((*2).quad)
48           where quad x = x*x
49
50 -- Loesung mit $ Operator
51 -- mapListe2 = map ((*2) $ quad)
52 --           where quad x = x*x
53
54 -- Aufgabe: Vektoraddition
55
56 -- todo
```