

## Übungsblatt 15

**Hinweis:** Programmieraufgaben immer per Email (eine Email pro Blatt und Gruppe) an den zuständigen Tutor schicken (Java Quellcode und eventuell benötigte Datendateien). Bitte werfen Sie Ihre schriftlichen Lösungen in die Briefkästen in Geb. 051, Erdgeschoss ein. Für den Erhalt von Bonuspunkten müssen Sie in wenigstens 10 Übungen anwesend sein.

### Aufgabe 1

Betrachten Sie die folgenden Haskell Funktionen:

```
inc n = n + 1  
dec n = n - 1
```

- Verwenden Sie diese beiden Funktionen, um die Funktion `add x y` zu implementieren. Diese Funktion addiert zwei natürliche Zahlen.
- Benutzen Sie die Funktion `add`, um die Funktion `mul x y` zu implementieren. Diese Funktion multipliziert zwei natürliche Zahlen.
- Benutzen Sie die Funktion `mul`, um die Funktion `fact n` zu implementieren. Diese Funktion berechnet die Fakultät einer natürlichen Zahl.

### Aufgabe 2

Implementieren Sie folgende Haskell Funktionen:

- `sortInsert`: Diese Funktion fügt ein Element an der richtigen Stelle in eine bereits sortierte Liste ein.

```
sortInsert 3 [1,2,4,5]  
[1,2,3,4,5]
```

- `removeElementAt`: Diese Funktion entfernt das Element an der angegebenen Position aus der Liste.

```
removeElementAt 0 [5,2,17,21,13]  
[2,17,21,13]
```

- `maxList`: Diese Funktion liefert das Element mit dem größten Wert zurück.

```
maxList [5,2,17,21,13]  
21
```

### Aufgabe 3

Betrachten Sie das folgende Haskell-Programm  $f$ , das für  $\mathbb{N}_0$  definiert ist ( $\mathbb{N}_0$  sind die natürlichen Zahlen inklusive 0). Was berechnet dieses Programm?

```
f 0 = g [0]
f 1 = g [1]
f x = g [y | y <- [2..x-1], mod x y == 0]

g [] = True
g (x:xs) = False
```

### Aufgabe 4

Betrachten Sie die folgenden Funktionen und berechnen Sie den Wert des Ausdrucks `take 1 (filter p [2..])`. Geben Sie alle Reduktionsschritte an.

```
take 0 xs          = []
take (n+1) []      = []
take (n+1) (x:xs) = x:take (n) xs

filter p []        = []
filter p (x:xs) = if (p x) then (x:filter p xs)
                  else (filter p xs)

p x = if mod x 2 == 0 then True
      else False
```