

(FP-Aufgaben 02)

Aufgabe 1: Welchen nicht-polymorphen Typ haben die folgenden Funktionen?

- a) $f1 \ x = 3.14 + x$
- b) $f2 \ x \mid x == 'A' = 0$

Aufgabe 2: Schreiben Sie die folgende Java-Funktion in eine linear-rekursive Haskell-Funktion um:

```
int potenz (int basis, int exponent) {  
    int ergebnis = 1;  
    while (exponent > 0) {  
        ergebnis = ergebnis * basis;  
        exponent = exponent - 1;  
    }  
    return ergebnis;  
}
```

Aufgabe 3: Sie haben x Cent und sehen ein Regal mit Bonbons, die 10 Cent, 20 Cent, 30 Cent usw. bis hinauf zu einem Euro kosten (x ist eine ganze positive Zahl).

Sie kaufen von jeder Sorte nur ein Bonbon, beginnend mit dem Bonbon für 10 Cent, bis Ihr Restgeld für ein weiteres Bonbon nicht mehr ausreicht. Schreiben Sie eine linear-rekursive Funktion, die rechnet, wie viele Bonbons Sie kaufen können.

Aufgabe 4: Schreiben Sie ein Programm, das den Mittelwert aller ganzen Zahlen von **x** bis **y** rechnet (**x** und **y** sind ganze positive Zahlen und dürfen in beliebige Reihenfolge angegeben werden).

Hinweis:

- In Haskell gibt es keine automatische Typ-Umwandlung.
- Falls Sie den Operator / auf ganze Zahlen anwenden möchten, verwenden Sie die Typkonverter-Funktion **fromIntegral**.

z.B. $\text{fromIntegral}(\text{summe } x \ y) / \text{fromIntegral}(\text{anzahl } x \ y)$