

## (FP-Aufgaben 10)

**Aufgabe 1:** Das folgende Programm ist fehlerhaft. Warum?

```
data Zahl = Eins | Zwei | Drei  
  
plus Eins Eins = Zwei  
plus Zwei Eins = Drei  
plus _ _ = Eins  
  
f :: [Zahl] -> Zahl -> Zahl  
f z y = if z == [] then y else (x + y)  
  where x :: Zahl  
        x = head z
```

**Aufgabe 2:**

- a) Definieren Sie einen Typ für Binäräume.

```
data Binaerbaum a = --TODO
```

- b) Definieren Sie eine Funktion, die kontrolliert, ob ein eingegebenes Element in dem Binärbaum existiert.

```
existiert :: a -> Binaerbaum a -> Bool
```

**Aufgabe 3:**

- a) Definieren Sie einen Typ für Bäume. Jeder Knoten des Baumes kann beliebige Anzahl von Unterbäumen enthalten.

```
data Baum a = --TODO
```

- b) Definieren Sie eine Funktion, die einen Baum als Parameter nimmt und eine Liste zurückliefert. Die Liste enthält alle Werte, die in dem Baum gespeichert sind.

```
baumZuListe :: Baum a -> [a]
```

**Aufgabe 4:** Eine Versicherungsfirma möchte ein neues System zur Verwaltung von Versicherungen entwickeln:

- Jede **Versicherung** hat eine Laufzeit, einen Beitrag, die Versicherungsscheinnummer und den Versicherungsvertreter, der den Versicherungsvertrag vermittelt hat.
- Die verschiedenen Versicherungen haben weitere Eigenschaften:
  - o Eine **Hausratsversicherung** hat eine Versicherungssumme und eine Angabe über die Quadratmeter der zu versichernde Wohnung.
  - o In einer **Haustierversicherung** ist die Anzahl der versicherten Haustiere angegeben.
- Ein **Versicherungsvertreter** hat einen Namen und eine Vermittlernummer.

- a) Definieren Sie sinnvolle Datentypen.

```
data Versicherung a = --TODO
```

- b) Schreiben Sie eine Funktion, die eine Liste von Versicherungstypen und eine Versicherungsscheinnummer als Parameter nimmt, nach der Scheinnummer sucht und den Namen des Vertreters zurückliefert.