

## FP-Aufgaben 09

### Aufgabe 1

Das folgende Programm ist fehlerhaft. Warum?

```
data A = B | C
class K a where
  f :: a -> Bool
  f _ = False
instance K A where
  f :: a -> Bool
  f x = x == C
  g x = x

aufruf = f B
```

- `g` ist nicht definiert in `class K`
- Typ Signatur "`f :: a -> Bool`" ist in **instance** deklaration nicht erlaubt
- **(==)** Funktion ist für **A** nicht definiert, wir können **deriving (Eq)** auf `data A` anwenden

## Aufgabe 2

Die folgende Klasse ist schon implementiert

```
class ListenGenerator a where
  listeGenerieren :: a -> [String]
```

Definieren Sie eine Instanz der Klasse ListenGenerator für Binärbäume.

```
data WochenTag
  = Montag
  | Dienstag
  | Mittwoch
  | Donnerstag
  | Freitag
  | Samstag
  | Sonntag
  deriving (Show)

data BinaryTree v
  = Empty
  | Branch v (BinaryTree v) (BinaryTree v)

class ListenGenerator a where
  listeGenerieren :: a -> [String]

instance Show v => ListenGenerator (BinaryTree v) where
  listeGenerieren Empty = []
  listeGenerieren (Branch val leftTree rightTree) =
    (listeGenerieren leftTree)
    ++ [show val]
    ++ (listeGenerieren rightTree)
```