Praktische Informatik 1 WiSe 98/99

5. Arbeitsblatt

Ausgabe: 14.1.1999
Bearbeitungszeit: zwei Tutorien

Prof. Bernd Krieg-Brückner
Shi Hui
Kolyang
Axel Lankenau
Christoph Lüth
Lutz Zegartowski

Lektüre

Auf diesem Arbeitsblatt werden Typklassen behandelt. Lest dazu bitte Kapitel 8 im Haskell–Buch.

Leitfragen

1. Was ist eine (Typ–)Klasse in Haskell?
2. Was ist der Unterschied zwischen einer Klasse (`class`) und einer Instanz (`instance`)?
3. In welcher Beziehung stehen Typen und Klassen in Haskell?
4. Wie können (Type–)Constraints in Haskell notiert werden?
5. Welche vordefinierten Typklassen sind Dir bekannt?
6. Wie lassen sich für algebraische Typen Instantiierungen für `Eq` ableiten?

Anwendungsaufgaben

1. Type–Constraints
   - Definiert eine Funktion `numEqual`, die zählt, wie oft eine Zahl `x` in einer Liste `xs` von `Int`s vorkommt. Definiert eine weitere Funktion `numLess` die zählt, wieviele Elemente in `xs` kleiner als `x` sind.
   - Erweitert die beiden Funktionen für Listen beliebigen Typs. Welche Eigenschaft müssen die Elementtypen besitzen? Welche Rolle spielen dabei Typklassen?
2. Class `Visible`
   Im Haskell–Buch wird die Klasse `Visible` wie folgt definiert:

```
class Visible t where
  toString :: t -> String
  size     :: t -> Int
```

   - Wie und unter welchen Voraussetzungen können Wahrheitswerte (Typ `Bool`), Paare (Typ `(t,u)`) oder Tripel (Typ `(t,u,v)`) zu `Visible`–Typen gemacht werden?
   - Definiere einen Enumerationstyp `Monat`, der als Werte die Kürzel der Monatsbezeichnungen (also `Jan, Feb` usw.) hat. Bilde eine Instanz der `Visible`–Klasse für den Monatstyp. Dabei soll die `size`–Funktion die Anzahl der Tage im entsprechenden Monat liefern, während die `toString`–Funktion den vollständigen Monatsnamen als String liefert.
   - Unter welchen Bedingungen ließe sich eine Queue, die Instanz von `Visible` ist, realisieren?