

**Aufgabe 1:****Klassen I**

(5 + 3 + 2 Punkte)

Wir betrachten eine nach den Haupthimmelsrichtungen ausgerichtete, gitterförmige Wegenetz einer Stadt. Alle 100 in Ost–West–Richtung verlaufenden Wege werden als Straßen bezeichnet, alle 100 in Nord–Süd–Richtung verlaufenden Wege als Alleen. Die Straßen und Alleen haben keine Namen sondern sind (übereinstimmend mit dem üblichen Koordinatensystem) nummeriert: Die südlichste Straße wird als 1. Straße bezeichnet, die westlichste Allee ist die 1. Allee und die Nummerierung der Straßen (bzw. Alleen) steigt nach Norden (bzw. Osten) hin an. Um Bewegungen zwischen zwei Kreuzungspunkten in diesem System zu beschreiben, verwendet man die folgenden Datentypen:

```
data Richtung = Nord | Ost | Sued | West
data Weg = Start Int Int | Next Weg Richtung
```

Der Konstruktor `Start` legt eine Kreuzung als Startpunkt an und mit dem Konstruktor `Next` wird die jeweils nächste Kreuzung in der angegebenen Himmelsrichtung angesteuert. Bei allen Bewegungen, die eine Koordinate außerhalb des Bereichs  $[1, 100]$  erzeugen (z.B. eine Bewegung von der 100. Straße nach Norden oder von der 1. Allee nach Westen) führen in einen Aus–Zustand von dem es kein zurück in das Wegenetz gibt (das Auto ist in den Fluss gefallen).

a) Implementieren Sie den Typ `Weg` als Exemplar der Klasse `Eq` wobei zwei Wege gleich sein sollen, wenn sie den gleichen Start und den gleichen Endpunkt haben, unabhängig von dem tatsächlich gelaufenen Weg. Alle Aus–Endpunkte sind als gleich zu betrachten.

b) Implementieren Sie den Typ `Weg` als Exemplar der Klasse `Show`, wobei der von `show` erzeugte String die folgende Gestalt haben sollte:

"Weg zur Kreuzung i.-te Allee, j.-te Strasse: k Alleen oestlich/westlich und n Strassen suedlich/noerdlich vom Startpunkt" oder "Weg ins Aus".

c) Implementieren Sie den Typ `Weg` als Exemplar der Klasse `Ord`, wobei zum Vergleich von zwei Wegen die folgenden Kriterien (mit fallender Priorität) herangezogen werden sollen:  $x$ -Koordinate (Allee) des Zielpunkts,  $y$ -Koordinate (Straße) des Zielpunkts,  $x$ -Koordinate des Startpunkts,  $y$ -Koordinate des Startpunkts. Die Koordinaten aller Aus–Punkte werden als  $(0, 0)$  angenommen.

**Aufgabe 2:****Klassen II**

(5 Punkte)

Wir betrachten Bäume in deren Knoten entweder Blätter sind oder innere Knoten mit genau zwei Kinderknoten, bei denen aber nicht zwischen linkem und rechtem Kind unterschieden wird. Außerdem wird jeder Knoten mit einer Farbe markiert. Zwei Blätter sind gleich, wenn sie die gleiche Farbe haben und zwei Bäume mit inneren Knoten sind gleich, wenn die Wurzeln die gleiche Farbe haben und die von den Kindern der Wurzeln definierten Unterbäume zu zwei Paaren von jeweils gleichen Unterbäumen zusammengefasst werden können. Wir verwenden die folgenden Datentypen

```
data Farbe = Rot | Blau | Gruen | Gelb | Schwarz
data Baum = Blatt Farbe | Knoten Farbe Baum Baum
```

a) Implementieren Sie den Typ `Baum` als Exemplar der Klasse `Eq` mit der oben beschriebenen Gleichheitsrelation

b) Implementieren Sie den Typ `Baum` als Exemplar der Klasse `Ord`, wobei  $t_1$  kleiner oder gleich  $t_2$  sein soll, wenn in  $t_2$  ein Knoten  $v$  existiert, so dass der von  $v$  definierte Unterbaum in  $t_2$  gleich  $t_1$  ist.

**Aufgabe 3: Decodierung von Präfixcodes (4 Punkte)**

Gegeben sei ein Präfixcode durch einen echten Binärbaum vom Typ

```
Tree = Leaf Char | Node Tree Tree
```

d.h. jedes Blatt ist mit einem Symbol markiert. Implementieren Sie eine Methode `decode :: String -> Tree -> String` die für einen Bit-String  $s$  einen Quellcode-String  $q$  berechnet, dessen Codierung  $s$  ergibt (sofern solch ein  $q$  existiert und sonst eine Fehlermeldung gibt).