SoSe 2014

Prof. Dr. Margarita Esponda

ProInformatik II: Funktionale Programmierung

1. Übungsblatt

1. Aufgabe

Schreiben Sie eine Haskell-Funktion ***teil***, die überprüft, ob eine Zahl genau von einer zweiten Zahl geteilt werden kann.

2. Aufgabe

Betrachten Sie folgende Haskell-Funktionsdefinition, die überprüfen soll, ob die eingegebene Zahl **n** eine ungerade Zahl ist.

```
ungerade :: Integer -> Bool
ungerade n = rem n 2 == 1
```

Ist die Funktion korrekt? Begründen Sie Ihre Antwort. Wenn Sie der Meinung sind, dass die Funktion inkorrekt ist, geben Sie eine korrekte Funktionsdefinition.

3. Aufgabe

Wenn die Seitenlängen eines rechtwinkligen Dreiecks **natürliche Zahlen** sind, werden diese Zahlen als **pythagoräische Zahlentripel** bezeichnet.

Definieren Sie eine Funktion in Haskell, die bei Eingabe dreier natürlicher Zahlen feststellen kann, ob es sich um die Seitenlängen eines rechtwinkligen Dreiecks handelt oder nicht. Mit anderen Worten soll die Funktion testen, ob die eingegebenen natürlichen Zahlen pythagoräische Zahlentripel sind.

4. Aufgabe

Schreiben Sie eine Funktion, die bei Eingabe von drei Zahlen **a**, **b** und **c** entscheiden kann, ob eine der drei Zahlen ein Mehrfaches der anderen zwei Zahlen ist.

5. Aufgabe

Schreiben Sie eine Funktion **toLower**, die beliebige Buchstaben in Kleinbuchstaben umwandelt.

6. Aufgabe

Schreiben Sie eine Funktion, die ein Zeichen als Argument bekommt und entscheiden kann, ob das Zeichen ein Klammerzeichen ist.

7. Aufgabe

Definieren Sie eine Funktion **weekday** in Haskell, die bei Eingabe eines Datums (in Form von drei positiven Zahlen) den Wochentag-Namen mit Hilfe folgender Formeln des Gregorianischen

Kalenders berechnet. Die Formeln berechnen eine Zahl zwischen 0 (Sonntag) und 6 (Samstag).

$$y_0 = year - \frac{14 - month}{12}$$

$$x = y_0 + \frac{y_0}{4} - \frac{y_0}{100} + \frac{y_0}{400}$$

$$m_0 = month + 12\left(\frac{14 - month}{12}\right) - 2$$

$$Name = \mathrm{mod}\left(\left(day + x + \frac{(31 \cdot m_0)}{12}\right), 7\right)$$

Ihre Funktionsdefinition soll folgende Signatur haben:

weekday :: Int -> Int -> Int -> String

Sie müssen in Ihrer Funktion kontrollieren, dass der Wertbereich der eingegebenen Zahlen korrekt ist und mit Hilfe der **error**-Funktion entsprechende Fehlermeldungen ausgeben.

Verwenden Sie eine **case**-Anweisung, um die Werte in Text zu verwandeln.

Anwendungsbeispiel:

```
weekday  20  10  2013   =>  "Sonntag"
```

a) Definieren Sie zuerst die Funktion mit Hilfe von **where**-Anweisungen.

b) Verwenden Sie in Ihrer Funktion **let**-Anweisungen.

8. Aufgabe

Die Fläche eines beliebigen regulären Polygons kann bei Eingabe der Seitenlängen **s** und der Anzahl der Seiten **n** mit Hilfe folgender Formeln berechnet werden.

$$area = \frac{nsa}{2}$$

mit *n = Anzahl der Seiten des Polygons*

s = Seitenlänge

$$a = Apothema = \frac{S}{2 \cdot \tan\left(\frac{\pi}{n}\right)}$$

Schreiben Sie entsprechende Haskell-Funktion, die die Berechnung macht. Verwenden Sie lokale Funktionen (mit Hilfe der **where**-Anweisung) für die Berechnung der Teilformeln.

9. Aufgabe

Nehmen Sie an, wir haben folgende Datentyp-Synonyme definiert:

```
type Point  = (Double, Double)
type Rectangle = (Point, Point)
```

Definiere unter Verwendung des **Rectangle**-Datentyps folgende Funktionen:

```
area :: Rectangle -> Double
overlaps :: Rectangle -> Rectangle -> Bool   -- Testet, ob die Rechtecke sich überlappen
contains :: Rectangle -> Rectangle -> Bool   -- Testet, ob eines der Rechtecke das
                                                andere Rechteck beinhaltet
```

Wichtige Hinweise:

1) Verwenden Sie geeignete Namen für Ihre Variablen und Funktionsnamen, die den semantischen Inhalt der Variablen oder die Semantik der Funktionen wiedergeben.

2) Verwenden Sie vorgegebene Funktionsnamen, falls diese angegeben werden.

3) Kommentieren Sie Ihr Programme.

4) Verwenden Sie geeignete lokale Funktionen und Hilfsfunktionen in Ihren Funktionsdefinitionen.

5) Schreiben Sie in allen Funktionen die entsprechende Signatur.