Praktische Informatik 3 WS 14/15

1. Übungsblatt

Ausgabe: 21.10.14
Abgabe: 31.10.14

Christoph Lüth
Sandor Herms
Daniel Müller
Jan Radtke
Henrik Reichmann
Sören Schulze
Felix Thielke

1.1 *Wellen und Schwingungen* *10 Punkte*

Trigonometrische Funktionen sind etwas Praktisches, wenn man zum Beispiel Rotationswinkel berechnen will. Im Standard-Prelude sind zwar trigonometrische Funktionen vordefiniert, aber diese benutzen natürlich Fließkommazahlen; das hat unter anderem das Problem der beschränkten Genauigkeit.

Deshalb wollen wir in diesem kurzen Beispiel eine Funktion implementieren, die den Kosinus für rationale Zahlen (`Rational`) beliebig genau approximiert. Dies erfolgt durch die Reihenentwicklung nach Taylor:

$$\cos x = \frac{x^0}{0!} - \frac{x^2}{2!} + \frac{x^4}{4!} - \ldots (-1)^n \frac{x^{2n}}{(2n)!} + \ldots$$

Implementieren Sie eine Funktion

```
cosinus :: Rational-> Rational-> Rational
```

welche den Kosinus durch eine Reihenentwicklung approximiert. Das erste Argument gibt an, bis zu welcher Genauigkeit gerechnet werden soll. Da der Fehler durch den Betrag des n-ten Reihenglieds nach oben begrenzt ist, kann die Berechnung abgebrochen werden, sobald dieses kleiner als der angegebene Wert wird.

1.2 *Zahlen mit Format* *10 Punkte*

Jetzt können wir den Kosinus berechnen, aber die Ausgabe ist unbefriedigend, denn rationale Zahlen werden immer in der normierten Darstellung als Paar aus Zähler und Nenner ausgegeben (wobei wir unter "ausgeben" im folgenden die Umwandlung in einen String verstehen):

```
*Format> 7185 % (-3190)
(-1437) % 638
```

In dieser Aufgabe wollen wir rationale (und damit auch ganze) Zahlen formatiert ausgeben. Die Formatierung wird durch eine *Formatierungsvorschrift* gegeben:

- Für die Vorkommastellen kann eine Länge vorgegeben werden; enthält die Zahl weniger Vorkommastellen kann vorgegeben werden, ob mit `0` oder Leerzeichen aufgefüllt wird. Ist keine Länge vorgegeben, werden alle Vorkommastellen ausgegeben.
- Ferner muss angegeben werden, wieviele Nachkommastellen höchstens ausgegeben werden. Werden dabei Nachkommastellen abgeschnitten, wird dieses durch ein nachgestelltes $*$ angezeigt.

1. Definieren Sie zuerst einen Datentyp Format, der eine Formatierung repräsentiert.
2. Damit implementieren Sie eine Funktion

   format :: Format→ Rational→ String

   welche die rationale Zahl wie in der Formatierungsvorschrift vorgegeben formatiert.