Funktionale Programmierung (ALP 1), WS 2011/12 — 1. Übungsblatt

Test ab 24. Oktober 2011

Bei Funktionen ist *generell* eine geeignete Typdeklaration anzugeben.

1. Dreiecke (10 Punkte)

   Schreiben Sie eine Funktion zum Testen, ob x, y, z die Seitenlängen eines Dreiecks (mit positiver Fläche) sind. (Entartete Dreiecke, wo zum Beispiel zwei Ecken zusammenfallen, sind ausgeschlossen.)

2. Negative Argumente (10 Punkte)

   Die zweistelligen Funktionen `mod x y` und `div x y` bestimmen den Rest von x bei Division durch y, beziehungsweise den ganzzahligen Quotienden der Division unter Vernachlässigung des Restes.

   (a) Finden Sie durch Probieren heraus, was passiert, wenn für x und y auch negative Werte oder 0 eingesetzt werden.

   (b) Welche Beziehung gilt immer (mit wenigen Ausnahmen; mit welchen?) zwischen `x - mod x y` und `div x y`?

   (c) Fassen Sie Ihre Ergebnisse zu Aufgabe (a) in möglichst einfache Regeln.

3. Überlauf (10 Punkte)

   Im Gegensatz zum Typ `Integer` haben Größen vom Typ `Int` einen Wert von höchstens $2^{31} - 1 = 2.147{,}483.647$ beziehungsweise $2^{63} - 1 = 9{,}223.372{,}036.854{,}775.807$.

   (a) Was passiert, wenn bei einer Rechnung mit Größen vom Type `Int` diese Grenze überschritten wird?

   (b) Welches ist der kleinste Wert, der im Typ `Int` dargestellt werden kann?

4. Zinseszinsberechnung (10 Punkte)

   Die folgende Funktion bestimmte die Zinsen einer Anlage von `zinsfuß` %.

   ```
   zinsen kapital zinsfuß = kapital * zinsfuß * 0.01
   ```

   (a) Definieren Sie unter Zuhilfenahme von `zinsen` eine Funktion `endwert` (mit geeigneten Parametern), die den Wert der Anlage (Kapital+Zinsen) am Ende einer Zinsperiode bestimmt.

   (b) Definieren Sie eine Funktion `endwert2`, die den Wert nach zwei Zinsperioden berechnet, wenn die Zinsen am Ende der ersten Periode wieder angelegt werden.

   (c) Funktioniert es auch mit der folgenden Definition?

   ```
   zinsen kapital zinsfuß = kapital * zinsfuß / 100
   ```

5. Gleitkommarechnungen (10 Punkte)

   Vom mathematischen Standpunkt aus müsste die folgende Funktion immer 0 liefern:

   ```
   zero:: Float -> Float
   zero x = (1/x)*x - 1
   ```

   Wegen Rundungsfehlern ist dies nicht immer der Fall. Finden Sie Werte x, bei denen das Ergebnis von 0 verschieden ist. (Ändert sich das Ergebnis beim Übergang zu `Double`?)

6. (10 Punkte) Schreiben Sie eine Funktion von drei `Integer`-Parametern, die ausgibt, wie viele der Eingabeparameter *größer* als der Durchschnittswert sind.