

Real World Haskell

Blatt 1

Julian Fleischer, Alexander Steen

Montag, den 22.07.2013

Aufgabe 0 (Basics) Installieren Sie die Haskell-Plattform auf Ihrem Rechner.

Aufgabe 1 (Kommandozeilenparameter)

Schreiben Sie eine Funktion `args :: IO [(String, Maybe String)]`, die die Kommandozeilenparameter, mit denen das Programm gestartet wurde, als Liste von Name-Wert-Paaren zurückgibt.

Es sollen Parameter der Form `arg=val` (Parameter mit Wert) und `arg` (Parameter ohne Wert) unterstützt werden.

Testen Sie Ihre Funktion mit folgendem Programm:

```
main = do
  argumente :: [(String, Maybe String)]
  argumente <- args
  print argumente
```

Aufgabe 2 (Ausdrücke auswerten) Betrachten Sie folgende Grammatik für einfache arithmetische Ausdrücke:

```
Expr := Expr Op Expr | nicht-negativer Float
Op   := + | - | * | /
```

Schreiben Sie eine Funktion `calculate :: String -> Float`, die den arithmetischen Wert einer Zeichenkette der obigen Sprache berechnet. Achten Sie auf Punkt-vor-Strich-Rechnung, aber vorerst nicht auf Fehlerbehandlung (z.B. Division-by-zero).

Sie können die Funktion `tokenize :: String -> [String]` von der Veranstaltungsseite verwenden.

Aufgabe 3 (Taschenrechner) Erstellen Sie ein ausführbares Haskell-Programm `calc`, welches den Wert eines als Kommandozeilenparameter übergebenen arithmetischen Ausdruck (aus Aufgabe 2) ausgibt. Sie können den Code der `main`-Funktion aus Aufgabe 1 entsprechend anpassen bzw. erweitern!

Beispiel: `./calc eq=3+5*4` liefert 23 als Ausgabe.

Aufgabe 4 (Interaktiver Taschenrechner) Erweitern Sie den Taschenrechner aus Aufgabe 3 folgendermaßen: Wird kein Parameter übergeben, so wird wiederholt auf Eingaben auf `stdin` gewartet und diese ausgewertet. *Optional:* Der Taschenrechner soll sich bei Eingabe von `exit` beenden.

Aufgabe 5 (Letzte Antwort) Erweitern Sie Ihren Taschenrechner aus Aufgabe 4 um die Möglichkeit, im interaktivem Betrieb bei Eingabe von `ans` den letzten berechneten Wert zurückzugeben. Schaffen Sie es, `ans` auch als Teil von Eingaben für nachfolgende Berechnungen zu erlauben? Denkbar wären Eingaben der Form `3+ans`, `ans*ans`, usw.

Aufgabe 6 (Dokumentation) Erstellen Sie eine *haddock*-Dokumentation Ihres Codes.