

Funktionale Programmierung- Aufgaben zur Einführung

Aufgabe 1

Starten Sie das Programm "winhugs". Untersuchen Sie die Effekte der untenstehenden Befehlszeilen (eventuelle Leerzeichen müssen bleiben!). Notieren Sie mögliche Bedeutungen der einzelnen Teile der Ausdrücke.

- `13+42`
- `2^10`
- `(+) 16 26`
- `map (*2) [1,2,3]`
- `filter (>5) [8,4,1,6,10]`

Aufgabe 2

Erstellen Sie eine Textdatei, in der Sie folgende Zeilen speichern:

```
square :: Int -> Int
square x = x^2
```

Speichern Sie die Datei unter dem Namen "first.hs".

Laden Sie die Datei in hugs mit

```
:l first.hs
```

Testen Sie die definierte Funktion mit

```
square 4
```

und weiteren selbstgewählten Eingaben. Wenn Fehler auftreten, welcher Art können diese sein? Lassen sich die Fehler analysieren und verbessern? Lässt sich `square` auch anders definieren?

Allgemeiner: Stellen Sie Vermutungen an, wie eine Funktion in der Programmiersprache Haskell definiert wird und welche Teile der Funktionsdefinition welche Bedeutung haben.

Aufgabe 3

Erweitern Sie "first.hs" um folgende Zeilen.

Hinweis: der senkrechte Strich | ist auf den meisten Tastaturen ein Sonderzeichen über der „7“. Je nach Betriebssystem muss eine zusätzliche Auswahl Taste gedrückt werden, meist „ALT“ oder „ALTGR“ oder ähnlich

```
strange :: Int -> Int
strange x
  | x > 10 = div (x+3) 2
  | x < 0  = (div (x*x) 3) +5
  | otherwise = square x
```

Testen Sie die neue Datei in hugs (aktualisieren mit ":r"). Untersuchen Sie die Bedeutung des **Pipe**-Zeichens (|) und des **Schlüsselwortes** "otherwise".