
Funktionale Programmierung

<http://proglang.informatik.uni-freiburg.de/teaching/functional-programming/2013/>

Übungsblatt 0 (Installation)

Di, 2013-10-22

Edit (2013-10-25) Auf Mac OS müssen *vor* der Installation der Haskell-Platform die „Command Line Tools for XCode“ installiert sein (oder alternativ auch das komplette XCode an sich).

<https://developer.apple.com/downloads/index.action>

Dafür muss man sich (leider) als Apple Developer (kostenlos) registrieren.

1 Download und Installation

Auf den Rechnern in den Mensa-Pools 21 und 29 ist die „Haskell-Platform“ (das bezeichnet den Compiler und die Standardbibliothek) bereits installiert.

Wenn Sie Windows oder Mac OS benutzen, laden Sie sich die „Haskell-Platform“ Version 2012.2.0.0 von

<http://www.haskell.org/platform/prior.html>

herunter und installieren Sie sie.

Wenn Sie eine Linux Distribution verwenden, dann versuchen Sie ein entsprechendes „Haskell-Platform“ Paket über den Paketmanager zu installieren (siehe auch

<http://www.haskell.org/platform/linux.html>.

Es ist jedoch möglich, dass die exakte Version 2012.2.0.0 nicht verfügbar ist. Es besteht die (geringe) Wahrscheinlichkeit, dass dadurch später Kompatibilitätsprobleme auftreten; dann muss zur Not die Installation im Pool benutzt werden.

Zum Editieren von Haskell-Quellcode kann ein gewöhnlicher Editor verwendet werden. Eine Übersicht von Editoren mit expliziter Haskell Unterstützung:

<http://www.haskell.org/haskellwiki/Editors>

2 Testen der Installation

(Der folgende Code nimmt Dinge aus der nächsten Vorlesung vorweg... Sie können also alternativ einfach mit dem Code aus der letzten Vorlesung testen)

Öffnen Sie eine Konsole und führen sie das Kommando `ghci` aus. Sie sollten nun den Prompt des GHC Interpreters sehen.

```
Prelude>
```

In GHCi können Sie (u.a.) Haskell-Quelldateien und Bibliotheken laden, definierte Werte inspizieren und Haskell-Einzeiler ausführen. GHCi versteht außerdem spezielle Kommandos, die mit einem Doppelpunkt (:) beginnen. Einige Beispiele:

Befehl

:q	beendet GHCi
:h	zeigt eine Übersicht aller GHCi Kommandos an
:l <.hs-Datei>	lädt eine Haskell Quelldatei
:r	lädt die letzte Quelldatei nochmals
:t <Ausdruck>	gibt den Typ eines Ausdrucks an
:cd <Verzeichnis>	wechselt das Arbeitsverzeichnis
:! <Shell Befehl>	führt beliebige Shell Kommandos aus

Inspizieren Sie den Wert „Hallo Welt!“:

```
Prelude> "Hallo Welt!"
"Hallo Welt"
Prelude> :t "Hallo Welt!"
"Hallo Welt" :: [Char]
```

Öffnen Sie eine Datei `Test.hs` mit einem Texteditor ihrer Wahl. Definieren Sie eine Begrüßungsfunktion:

```
1 greetings :: String -> String
2 greetings name = "Hallo " ++ name ++ "!"
```

Laden Sie `Test.hs` in GHCi und grüßen Sie jemanden

```
Prelude> :l Pfad/zu/Test.hs
*Main> greetings "Welt"
"Hallo Welt!"
```