

Arbeitsblatt: Listen

1. Listen als Modell

Wie erwähnt sind Listen die zentrale Datenstruktur in der funktionalen Programmierung. Listen können verwendet werden um diverse unterschiedliche Datensätze zu modellieren. Gesucht sind die Typen für die beschriebenen Daten sowie ein Beispiel. Verwenden Sie Kombinationen von Listen, Tuples und den Basic Types.

a. *Börsenkurse*

Aktiennamen und deren Dollar Kurse.

Type: [(String, Float)]

Beispiel: [("GOOG", 875.92), ("MSFT", 32.85), ("ORCL", 33.75)]

b. *Einkaufsliste*

Eine Liste der zu kaufenden Artikel jeweils mit der benötigten Anzahl.

Type:

Beispiel:

c. *Stundenplan*

Für jeden Wochentag eine Abfolge der Module.

Type:

Beispiel:

d. *Videothek Datenbank*

Für jeden Kunde den Namen und alle ausgeliehenen Filme, jeweils mit der Information, ob der Film bereits wieder zurückgebracht wurde.

Type:

Beispiel:

2. Listen zerlegen

Gesucht Sie die Resultate der folgenden Ausdrücke:

Ausdruck	Resultat
<code>length (tail [1,2,3])</code>	<code>2 :: Int</code>
<code>head (head [1,2,3])</code>	
<code>head (tail (tail [1,2,3]))</code>	
<code>tail (tail [1,2,3])</code>	
<code>tail (tail (tail [1,2,3]))</code>	
<code>tail (tail (tail (tail [1,2,3])))</code>	
<code>length (tail (tail (tail [1,2,3])))</code>	

3. Wichtige Funktionen über Listen

Gesucht sind die wichtigsten Funktionen auf Listen. Auf dem AD finden Sie ein vorbereitetes Excel Dokument, indem Sie pro Funktion jeweils die fehlenden Felder einfüllen können.

Verwenden Sie <http://www.haskell.org/hoogle> und GHCi um diese Aufgabe zu lösen.