



Technische Universität Berlin

Software and Embedded Systems Engineering Group

Prof. Dr. Sabine Glesner

www.sese.tu-berlin.de    Sekr. TEL 12-4    Ernst-Reuter-Platz 7    10587 Berlin





# Softwaretechnik und Programmierparadigmen WiSe 2022/2023



Prof. Dr. Sabine Glesner  
Milko Monecke  
Simon Schwan

## Übungsblatt 04

### Aufgabe 1: Module und Importe



- a) Fasst alle in den letzten Tutorien entwickelten Teile unseres Taschenrechners als Stack-Modul mit dem Namen *Command* zusammen, in dem alles exportiert wird. 
- b) Nun sollen nach außen hin nicht alle Teile des Moduls sichtbar sein: Insbesondere `fee` geht niemanden etwas an. Außerdem dürfen `Commands` keine negativen Werte enthalten. Versteckt daher auch den Konstruktor `ValR` und stellt dafür eine Funktion `toCommand :: Int -> Command` zur Verfügung, die nur für positive Integers definiert ist. Wurde eine solche Funktion eigentlich schon implementiert? 

### Aufgabe 2: \$ and do





- a) Was macht der \$ Operator:  $(\$) :: (a \rightarrow b) \rightarrow a \rightarrow b$ ? 
- b) Erarbeitet ein Programm, in dem ein Name als Eingabe gefordert wird und anschließend `Hello {name}` auf der Konsole ausgegeben wird mit der `do` Notation. 

---

#### Schlüssel:

-  Ein ergänzendes Video wird zur Vor- oder Nachbereitung veröffentlicht.
-  Wird im Tutorium besprochen.

### Aufgabe 3: Testen in Haskell

- a) Was sind Unit Tests? 
- b) Implementiert Unit-Tests für die Funktionen `toCommand` und `eval` mithilfe der *Hspec*-Bibliothek. Die Tests sollen mit dem Befehl `stack test` ausgewertet werden. 
- c) Teilt die Testfälle zur Übersicht in verschiedene Test-Spezifikationen auf. 
- d) Wurden beide Funktionen ausreichend getestet? Überprüft eure Testfälle mithilfe der *Haskell Program Coverage* (HPC) und erweitert ggf. eure Tests, um beide Funktionen vollständig zu überprüfen.   
Hinweis: Nutzt die Befehle `stack clean` und `stack test --coverage`.