

---

**Arbeitspaket zu KW 22 zum Modul „Fortgeschrittene Programmierung“**  
Sommersemester 2020

---

Nach der gründlichen Vorbereitung in den vergangenen Wochen geht es nun um Funktionen höherer Ordnung, also Funktionen, die Funktionen als Argumente bekommen und zurückgeben. Da, wie wir aus dem Modul Modellierung wissen, Konstanten auch Funktionen sind, ist dies eine Erweiterung der Funktionen, die wir schon kennen.

Insbesondere durch die Funktion `fold` (und Spezialfällen wie `map` und `filter`) lassen sich z.B. Rekursionschemata, mit denen wir uns in den vergangenen Wochen ausführlich beschäftigt haben, sehr kompakt ausdrücken.

### Lesen und Verstehen

Kapitel 6 Funktionen höherer Ordnung

im Buch „Haskell-Intensivkurs - Ein kompakter Einstieg in die funktionale Programmierung“ (<https://link.springer.com/content/pdf/10.1007%2F978-3-642-04718-3.pdf> )

In diesem Kapitel wird fast ausschließlich mit Listen gearbeitet. Es ist aber sehr hilfreich, sich schon beim Lesen zu überlegen, wie `map` und `fold` für Peano-Zahlen und Bäume aussehen könnten.

6 Punkte für Zusammenfassungen, je einen für folgende Abschnitte mit einem weiteren Beispiel dazu, welches nicht im Buch steht:

- 6.1. `map`
- 6.2 `filter`
- 6.3 (und 6.4) `fold`
- 6.5 (und 6.6) `zip`
- 6.7 Verknüpfung von Funktionen
- 6.8 Currying

### Übungsaufgaben

Für Serie 6 gibt es insgesamt 9 Aufgaben-Punkte:

- 3 für 6.1, je einen für 6.1.(a,b,c), (d,e) und f
- 2 für 6.2, je einen für 6.2.a und 6.2.b
- 2 für 6.3, je einen für 6.3.a und 6.3.b
- 2 für 6.4, je einen für 6.4.(a,b) und (c,d)

### Autotool

Haskell-Funktionen mit Funktionen höherer Ordnung für Listen