

(Aufgaben-2)

Aufgabe 1: Schreiben Sie Funktionstypen für die Aufgaben des ersten Aufgabenblattes.

Aufgabe 2: Welchen Typ haben die folgenden Haskell-Ausdrücke?

- `[(1.3,'a')]`
- `[]`

Aufgabe 3: Ein häufig verwendeter Datenstruktur ist der *Stack*. Auf einem Stack werden Daten ausschließlich nach dem *LIFO*-Prinzip (*Last In First Out*) abgelegt.

Ein Element kann mittels der **push**-Operation oben auf den Stack gelegt werden, die **top**-Operation liefert das oberste Element des Stackes zurück, und die **pop**-Operation entfernt das oberste Element vom Stack.

Definieren sie das Stack-Konzept mit Hilfe Funktionen, die Sie für **push**, **top** und **pop** schreiben. Beachten Sie dabei, dass eine Hilfsfunktion **istLeer**, die testet, ob ein Stack leer ist, ist benötigt.

Hinweis:

- Die einfachste Datenstruktur für einen Stack ist eine Liste.
- Überlegen Sie welche Eigenschaften Ihre definierten Funktionen haben müssen.

Zum Beispiel:

der Aufruf **istLeer (push element stack)** sollte **False**,
der Aufruf **top (push element stack)** sollte **element** und
der Aufruf **pop (push element stack)** sollte **stack** zurückliefern.

Aufgabe 4: Schreiben Sie eine Funktion, die das n-te bis m-te Element einer Liste ausgibt.

`unterListe :: Int → Int → [a] → [a]`

Aufgabe 5: Schreiben Sie eine Funktion, die die ersten **n** Fibonacci-Zahlen in eine Liste aufammelt.

`fibListe :: Int → [Int]`