

**Präsenzübungen zur Vorlesung**  
**Deklarative Programmierung: Sommersemester 2018**  
Nr. 4

---

**Aufgabe 4.1:** Natürliche Zahlen

In der Vorlesung haben Sie gesehen, dass man natürliche Zahlen als Instanz rekursiver Datentypen ansehen kann. Implementieren Sie die folgenden Funktionen nur unter Verwendung von `zero?`, `positive?`, `add1`, `sub1`, `0`:

- a) `(nat-add m n)`, die die beiden Zahlen addiert.
- b) `(nat-sub m n)`, die die beiden Zahlen subtrahiert. Sollte  $m \leq n$  gelten, ist das Ergebnis 0.
- c) `(nat-mult m n)`, die die beiden Zahlen multipliziert.
- d) `(nat-pow m n)`, die  $m^n$  berechnet.

**Aufgabe 4.2:** Listige Rekursion

- a) Programmieren Sie eine rekursive Funktion, welche die Elemente einer Liste aufaddiert.
- b) Programmieren Sie eine rekursive Funktion, welche die restlichen Elemente vom ersten Element abzieht.
- c) Programmieren Sie eine rekursive Funktion, welche die erste Hälfte der Liste miteinander multipliziert und die Summe der zweiten Hälfte von der Multiplikation abzieht (bei ungerade Listen können Sie davon ausgehen, dass die erste Hälfte ein Element mehr umfasst).  
Bsp.: `(recMulSub (list 4 3 2 1))`  $\rightarrow 9$

**Aufgabe 4.3:** Listige Funktionen

Implementieren Sie die folgenden Funktionen für Listen ganzer Zahlen:

- a) `(list-size l)`, die die Länge der Liste berechnet.
- b) `(list-contains e l)`, die überprüft, ob sich `e` in der Liste `l` befindet.
- c) `(list-max l)`, die das größte Element der Liste findet und zurückliefert.