

## Übungen Funktionale Programmierung (in Clojure) Serie 2

### 1. Abstrakter Syntaxbaum von Infix-Ausdrücken

Zeichnen Sie zu folgenden Ausdrücken den abstrakten Syntaxbaum:

- (a)  $2 + 3 + 1$
- (b)  $2 * 3 + 4$
- (c)  $2 * (3 + 4)$
- (d) `true and false or true`
- (e) `square(7) - 7`
- (f)  $(1 = 1) \text{ or } (1 \neq 1)$
- (g)  $(3 + 4) * (2 + 5)$

### 2. Ausdrücke in der Präfix-Notation

Schreiben Sie folgende Ausdrücke in der Präfix-Notation:

- (a)  $2 + 3 + 1$
- (b)  $2 * 3 + 4$
- (c)  $2 * (3 + 4)$
- (d) `true and false or true`
- (e) `square(7) - 7`
- (f)  $(1 = 1) \text{ or } (1 \neq 1)$
- (g)  $(3 + 4) * (2 + 5)$

Wie hängt dies mit der vorherigen Aufgabe zusammen?

### 3. Ausdrücke in der Repl

Was ergeben die folgenden Ausdrücke in der Repl, eingegeben in der vorgegebenen Reihenfolge. Bitte überlegen Sie zuerst im Kopf, was das Ergebnis sein muss und probieren Sie erst dann aus.

```
user=>10  
  
user=>(+ 5 3 4)  
  
user=>(- 9 1)  
  
user=>( / 6 2)  
  
user=>( / 7 2)  
  
user=>( / 8 6)  
  
user=>( + (* 2 4) (- 4 6))  
  
user=>(def a 3)
```

```
user=>(def b (+ a 1))

user=>(+ a b (* a b))

user=>(= a b)

user=>(if (and (> b a) (< b (* a b))) b a)

user=>(cond
  (= a 4) 6
  (= b 6) (+ 6 7 a)
  :else 25)
```

#### 4. Daten = Code

- (a) Wie kann man die Liste (+ 2 3) als eine Liste von Werten behandeln?
- (b) Und wie kann man eine Liste (als Daten) mit den Elementen +, 2 und 3 als Code ausführen?