

Programming Concepts and Paradigms (PCP)

Programmierübung zu Clojure Woche 1

Hauptthemen: Grundlagen, Funktionale Modellierung, Bedingte Ausdrücke, erste Rekursionen
Zeitfenster: ca. 4-6 Lektionen
Ingmar Baetge, Roger Diehl, Marcel Baumann

1. Aufgabe

Schreiben Sie in mathematischer Notation und berechnen Sie folgende Formen

```
(/ 1 (+ 1 1))  
(/ 1 (+ 1 (/ 1 (+ 1 1))))  
(/ 1 (+ 1 (/ 1 (+ 1 (/ 1 (+ 1 1))))))
```

Was fällt bei den Ergebnissen auf?

2. Aufgabe

- Definieren Sie Länge und Breite eines Rechtecks und berechnen Sie Fläche, Umfang und Länge der Diagonale als einfachen Ausdruck.
- Definieren Sie Länge, Breite und Höhe eines Quaders und berechnen Sie Volumen und Oberfläche als einfachen Ausdruck.
- Verpacken Sie die Berechnungen aus a) und b) in Funktionen (Länge, Breite und beim Quader auch die Höhe sind dann Parameter).

3. Aufgabe

Gegeben sind folgende *Clojure*-Ausdrücke:

```
(+ (* 2 3) (/ 3 4))  
(+ - * /)  
'(+ - * /)  
(str 1.0 2 true)  
(+ (parse-long "123") (parse-double "0.45"))  
(or "this" "is" "crazy")  
(and "this" "is" "crazy")  
(not "this" "is" "crazy")  
((fn [x] (/ 1 x)) (/ 5 2))  
(int? 3000000000)
```

Prüfen Sie für jeden Ausdruck, ob dieser gültig ist oder einen Fehler liefert. Erklären Sie jeweils, was der Ausdruck berechnet und welchen Datenwert (inklusive Typangabe) der Ausdruck zurückliefert, oder welcher Fehler vorliegt.

4. Aufgabe

Darf man in den folgenden Beispielen die Bedingungsklauseln vertauschen, ohne dass sich die Semantik des Programms ändert?

a) Temperaturen:

```
(cond
  (> temperatur 35) "heiss"
  (> temperatur 25) "warm"
  (> temperatur 15) "mittel"
  :else "kalt")
```

b) Teilbarkeit:

```
(cond
  (zero? (rem zahl 2)) "durch 2 teilbar"
  (zero? (rem zahl 3)) "durch 3 teilbar"
  :else "weder durch 2 noch durch 3 teilbar")
```

c) Wandeln Sie das Programm aus a) in einen Ausdruck um, der nur **if** verwendet.

d) Könnte man die Beispiele aus a) und b) auch mit **case** implementieren? Begründen Sie.

5. Aufgabe

Gegeben ist eine quadratische Gleichung $ax^2 + bx + c = 0$, die Lösung dieser Gleichung liefert bekanntlich die sogenannte Mitternachtsformel:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Tipp: Eine Hilfsfunktion, die den Teil unter der Wurzel berechnet (die *Diskriminante*) kann bei allen drei Teilaufgaben hilfreich sein.

- Schreiben Sie ein Prädikat, welches testet, ob eine Gleichung mit den Parametern a, b und c lösbar ist. Achten Sie auf eine gute Namensgebung.
- Schreiben Sie eine Funktion, die die Koeffizienten a, b und c als Parameter entgegennimmt und die Anzahl der Lösungen zurückliefert (0, 1 oder 2).
- Schreiben Sie eine Funktion, die einen Vektor ([...]) mit den Lösungen zurückliefert – dieser Vektor kann 0, 1 oder 2 Elemente haben.

6. Aufgabe

Gegeben sind folgende Definitionen:

```
(def a '(* 3 4))
(def b 'a)
(def c 'b)
```

Wenn Sie nur das Symbol `c` verwenden dürfen, wie könnten Sie trotzdem das Ergebnis der Berechnung erhalten? Tipp: Recherchieren Sie die Funktion `eval`.

7. Aufgabe

- Schreiben Sie eine Funktion, die CHF-Betrag in Euro umrechnet.
- Schreiben Sie eine Funktion, die einen Celsius-Wert in Fahrenheit umrechnet.
- Stellen Sie sich folgenden Funktionsaufruf vor, um die Temperatur 30° Celsius nach Fahrenheit umzurechnen:

```
(scale 0.0 100.0 32.0 212.0 30.0)
```

Die ersten beiden Parameter sind hier zwei Referenzwerte auf der Celsius-Skala, und danach folgen die beiden Werte in Fahrenheit, die diesen Celsius-Temperaturen entsprechen. Der letzte Wert ist der Wert, der von der ersten in die zweite Skala umgerechnet werden soll.

Wie könnte eine Funktion aussehen, die allgemeine lineare Umwandlungen zwischen Skalen berechnet? Wie könnte man damit auch Währungen umrechnen?

8. Aufgabe

- Schreiben Sie eine rekursive Funktion

```
(defn sum-range [a b] ...)
```

welche die ganzen Zahlen von `a` bis `b` aufsummiert (`b` inklusive). Zwei Beispielaufrufe:

```
(sum-range 1 5) ; ergibt 15
(sum-range 3 9) ; ergibt 42
```

Verwenden Sie ausschliesslich einfache Funktionen wie `inc`, `dec`, `+` und rekursive Aufrufe.

- Erweitern Sie die Funktion, so dass die Quadratzahlen von `a` bis `b` aufsummiert werden.

```
(sum-squares 1 4) ; 1^2 + 2^2 + 3^2 + 4^2 = 30
```

9. Aufgabe

Betrachten Sie folgenden Algorithmus:

Eingabe: natürliche Zahl n , $n > 1$.

```
Solange  $n > 1$ :  
   $n$  ausgeben  
  Wenn  $n$  gerade:  
    Setze  $n = n/2$   
  sonst:  
    Setze  $n = 3*n + 1$ 
```

Welche Zahlenfolgen werden ausgegeben, wenn Sie den Algorithmus mit $n=3$ oder $n=7$ ausführen?

Setzen Sie diesen Algorithmus rekursiv mit *Clojure* um!