

Übungen Funktionale Programmierung (in Clojure) Serie 9

1. Tiere polymorph

- (a) Programmieren Sie in Java eine abstrakte Klasse `Tier` mit einer Methode `spricht`. Bilden Sie Subklassen `Katze` und `Hund`, die bei der Methode `spricht` "Wauwau" bzw. "miauäusgeben".
Erzeugen Sie ein paar Objekte und lassen sie sprechen
- (b) Programmieren Sie dieselbe Funktionalität in Clojure mit Multimethoden.

2. `clojure.inspector`

- (a) Machen Sie sich mit der Bibliothek `clojure.inspector` vertraut und machen Sie einige Beispiele:

```
(inspect-tree '(and p (or q r)))
```



```
(inspect-tree [1 2 3 [4 5 6 [7 8 9]]])
```



```
(inspect-tree {:clojure {:rev "1.6.0" :java "SE 6"}})
```



```
(inspect-table [[:a :b :c] [1 2 3] ["a" "b" "c"]])
```
- (b) Analysieren Sie den Quellcode in `org.clojure.inspector.clj` und erläutern Sie, wie Multimethoden in diesem Beispiel eingesetzt werden.

3. Flächenberechnung mit Multimethoden

- (a) Programmieren Sie Funktion `make-rect` und `make-circle`, die Hash-Maps für Rechtecke und Kreise erzeugen.
- (b) Programmieren Sie eine Multimethode (`area ...`), die die Fläche eines Rechtecks oder eines Kreises berechnet.

4. Flächenberechnung mit Records und Protocols

- Lösen Sie die vorherige Aufgabe mittels (`defrecord ...`) und (`defprotocol ...`).