

Übungen Programmieren in Clojure Serie 4

1. Strings

Schreiben Sie Funktionen für folgende Aufgaben:

- (a) `(str-first s)` gibt den ersten Buchstaben des Strings `s` zurück.
- (b) `(str-last s)` gibt den letzten Buchstaben des Strings `s` zurück.
- (c) `(str-half s)` gibt die erste Hälfte des Strings `s` zurück.

2. Vektoren

Schreiben Sie folgende Funktionen:

- (a) `(vsum vec)` summiert alle Werte eines Vektors ganzer Zahlen auf.
- (b) `(vsum1+3 vec)` gibt die Summe des ersten und des dritten Elements des Vektors `vec` von ganzen Zahlen zurück.
- (c) `(take-3 vec)` gibt einen Vektor bestehend aus den ersten drei Elementen von `vec` zurück.
- (d) `(monotonic? vec)` ermittelt, ob ein Vektor ganze Zahlen diese in streng monoton aufsteigender oder absteigender Reihenfolge enthält.

3. Komplexe Zahlen

Entwickeln Sie einen Datentyp für komplexe Zahlen, den Sie als einen Vektor repräsentieren, bestehend aus Real- und Imaginärteil.

- (a) Schreiben Sie Funktionen `c+`, `c-`, `c*` und `cdiv` die zwei komplexe Zahlen addieren, subtrahieren, multiplizieren bzw. dividieren.
- (b) Erweitern Sie die Funktion `c+`, so dass man beliebig viele komplexe Zahlen addieren kann.
- (c) Schreiben Sie eine Funktion `c||`, die den Betrag einer komplexen Zahl ermittelt.
- (d) Schreiben Sie eine Funktion `polar`, die zu einer komplexen Zahl in unserer algebraischen Form die Repräsentation in Polarform ermittelt. Repräsentieren Sie die Polarform auch als Vektor mit zwei Elementen und verwenden Sie Metadaten, um diese Darstellung als Polarform zu „markieren“.

4. Maps

Für die folgende Aufgabe können Sie die Datei `books.edn` verwenden. `edn` steht für „Extensible Data Notation“ und ist ein Format für die Serialisierung von Clojure Datenstrukturen.

Man kann in Clojure den Inhalt der Datei an ein Symbol binden durch

```
(def books (edn/read-string (slurp "books.edn")))
```

- (a) Finden Sie heraus, wie die Werte in `books.edn` strukturiert sind.
- (b) Schreiben Sie eine Funktion, die zu einem Buch den Autor ermittelt.
- (c) Schreiben Sie eine Funktion, die zu einem Buch die Länge des Titels ermittelt.

- (d) Schreiben Sie eine Funktion `find-by-isbn`, die zu einer übergebenen ISBN das Buch findet.
- (e) Schreiben Sie eine Funktion `find-by-author`, die zu einem übergebenen Autor die Bücher dieses Autors findet.
- (f) Schreiben Sie eine Funktion (`find keyword suchstring`), die zu einem Suchstring alle Bücher findet, deren dem `keyword` entsprechendes Feld den Suchstring enthält.

5. Transients

Schreiben Sie eine Funktion (`freq coll`), die wie `frequencies` eine Map der verschiedenen Elemente einer Kollektion mit der Zahl ihres Vorkommens ermittelt. Verwenden Sie dazu *persistente* Datenstrukturen.

Vergleichen Sie dann die Laufzeit Ihrer Funktion mit `frequencies`.