

Übungen Funktionale Programmierung (in Clojure) Serie 7

1. Vektoren

Für die folgende Aufgabe verwenden Sie bitte die Funktionen, die wir bisher besprochen haben. Wir werden im nächsten Übungsblatt sehen, dass vieles auch viel eleganter geht.

- Schreiben Sie eine Funktion `vsquare`, die aus einem Vektor einen anderen Vektor erstellt, der die Quadrate der Elemente enthält.
- Schreiben Sie eine Funktion `vinc`, die aus einem Vektor einen anderen Vektor erstellt, dessen Elemente um 1 höher sind.
- Schreiben Sie eine Funktion `vmult`, die die Elemente eines Vektors mit einer Zahl multipliziert.
- Schreiben Sie eine Funktion, die aus zwei Vektoren gleicher Länge das Skalarprodukt berechnet.

2. Bücherliste als Vektor

- Erstellen Sie einen Vektor, der Hashmaps von Informationen über Bücher mit ISBN, Autor und Titel enthält.
- Schreiben Sie eine Funktion `search-author`, die in der Bücherliste nach Büchern eines Autors sucht und diese ausgibt.
- Schreiben Sie eine Funktion `get-author-set`, die die Menge der Autoren der Bücher ausgibt.
- Sortieren Sie die Menge der Autoren.

3. Rechtecke

Wir betrachten Rechtecke in der euklidischen Ebene, deren linke untere Ecke am Punkt $(0,0)$ liegt. Ein solches Rechteck wird durch die rechte obere Ecke (x,y) eindeutig bestimmt.

- Wie kann man Rechtecke in Clojure repräsentieren?
Schreiben Sie eine Funktion `(rectangle x y)`, die ein Rechteck „erzeugt“.
- Gegeben ein Rechteck `rectangle`, schreiben Sie Funktionen für die Bestimmung
 - der Höhe (`height rectangle`)
 - der Breite (`width rectangle`)
 - der Fläche (`area rectangle`)
- Schreiben Sie Funktionen für die Untersuchung von Rechtecken:
 - Ist es ein Quadrat? (`square? rectangle`)
 - Enthält es einen bestimmten Punkt? (`contains-point? rectangle point`).
Der Punkt sei repräsentiert durch einen Vektor mit x - und y -Koordinate.
 - Liegt ein Rechteck innerhalb eines anderen? (`contains-rectangle? outer inner`)