

Aufgaben – ausgewählte Lösungen

Einführung in die Programmierung

Johannes Brauer

15. Januar 2021

Inhaltsverzeichnis

| | | |
|----------|--|----------|
| 1 | Erste Schritte in Racket | 1 |
| 2 | Auswertung arithmetischer Ausdrücke | 1 |
| 3 | Aufschreiben elementarer Funktionen | 1 |
| 4 | Aufschreiben elementarer Funktionen | 2 |
| 5 | Anwenden der Aufschreiberegeln | 3 |
| 6 | Profit für den Kinobesitzer | 3 |

1 Erste Schritte in Racket

Machen Sie sich mit den NORDAKADEMIE-Rechnern vertraut und richten Sie Ihren Arbeitsplatz ein (Mail, Webbrowser, Verzeichnisse für die Vorlesungen usw.).

Finden und starten Sie DrRacket nach den Anweisungen in der Vorlesung. Werten Sie einen ersten Ausdruck aus, z. B. `(* 6 7)`.

Welche Funktion haben die Buttons? Welche Menü-Befehle verstehen Sie schon?

Schauen Sie sich in einem Webbrowser die Seiten zu Racket unter <https://racket-lang.org/> an. Wo finden Sie Hilfe zur Bedienung von DrRacket? Wie können Sie sich über die Sprache Racket informieren? Wo finden Sie alle vordefinierten mathematischen Funktionen?

2 Auswertung arithmetischer Ausdrücke

- Wie wird der Ausdruck
`(* (+ 2 2) (/ (* (+ 3 5) (/ 30 10)) 2))`
ausgewertet?
- Experimentieren Sie mit verschiedenen Operatoren und Zahlenarten.
- Werten Sie die folgenden Ausdrücke aus und vergleichen Sie die Resultate:

```
(- 1.0 0.9)
(- 1000.0 999.9)
(- #i1000.0 #i999.9)
```

3 Aufschreiben elementarer Funktionen

Schreiben Sie für die folgenden mathematischen Formeln Racket-Funktionsdefinitionen auf:

- $n^2 + 1$

```
(define fa
  (lambda [n]
    (+ (* n n) 1)))
```

b. $\frac{1}{2}n^2 + 3$

```
(define fb
  (lambda [n]
    (+ (/ (* n n) 2) 3)))
```

c. $2 - \frac{1}{n}$

```
(define fc
  (lambda [n]
    (- 2 (/ 1 n))))
```

Geben Sie die Racket-Funktionen in das Definitionsfenster von DrRacket ein. Geben Sie anschließend in das Interaktionsfenster Funktionsaufrufe für diese Funktionen ein.

4 Aufschreiben elementarer Funktionen

In der Praxis findet der Programmierer selten mathematische Formeln vor. Aufgabenstellungen sind eher als Prosatext gegeben. Die Berechnungsformeln muss er selbst entwickeln durch

- eigenes Nachdenken,
- Nachschlagen in geeigneten Quellen oder
- Nachfragen beim Auftraggeber.

Finden Sie für die folgenden Aufgabenstellungen die passenden Formeln und schreiben Sie diese als Funktionsdefinitionen in *Racket* auf:

- a. Berechnung des Rauminhalts eines Quaders aus dessen Länge, Breite und Höhe.

```
(define quader-volumen
  (lambda [laenge breite hoehe]
    (* laenge breite hoehe)))
```

- b. Schreiben Sie eine Funktion, die aus der Entfernung und den Geschwindigkeiten zweier Züge die Zeit ermittelt, nach der die Züge sich treffen, wenn Sie sich auf einem gemeinsamen Streckenabschnitt von ihren jeweiligen Startpunkten aus aufeinander zu bewegen.

```
(define zug-treffen
  (lambda [entfernung geschwndgkt1 geschwndgkt2]
    (/ entfernung (+ geschwndgkt1 geschwndgkt2))))
```

- c. Berechnung der Miete, die eine Spielerin in Monopoly bezahlen muss, falls sie auf einen Bahnhof trifft, der einer anderen Spielerin gehört. Die Miete ist davon abhängig wie viele Bahnhöfe der anderen Spielerin gehören:

| Anzahl der Bahnhöfe | Miete |
|---------------------|-------|
| 1 | 500 |
| 2 | 1000 |
| 3 | 2000 |
| 4 | 4000 |

Hinweis: Ein Aufruf `(expt x y)` liefert x^y als Ergebnis.

```
(define bahnhofsmiete
  (lambda [anzahl-bahnhoeefe]
    (* 500 (expt 2 (- anzahl-bahnhoeefe 1)))))
```

5 Anwenden der Aufschreiberegeln

Schreiben Sie die Funktion zur Berechnung der Bahnhofsmiete in Monopoly (s. o.) gemäß den Regeln 1 bis 3 aus der Vorlesung auf.

```
;; berechnet die Bahnhofsmiete aus der Anzahl der Bahnhöfe
(define bahnhofsmiete
  (lambda [anzahl-bahnhoeefe]
    (* 500 (expt 2 (- anzahl-bahnhoeefe 1)))))
;; Beispielanwendungen
(= (bahnhofsmiete 4) 4000)
(= (bahnhofsmiete 3) 2000)
(= (bahnhofsmiete 2) 1000)
(= (bahnhofsmiete 1) 500)
```

Wenn nichts anderes angegeben ist, sind auch die Funktionen für die folgenden Aufgaben gemäß diesen Regeln aufzuschreiben!!!

6 Profit für den Kinobesitzer

Ein altmodisches Vorstadtkino besitzt eine einfache Formel für die Berechnung des Profits einer Vorstellung: Jeder Kinobesucher bezahlt 500 Währungseinheiten für die Eintrittskarte. Jede Vorstellung kostet das Kino 2000 Währungseinheiten plus 50 Währungseinheiten pro Besucher. Schreiben Sie eine Funktion `profit` zur Berechnung des Profits bei gegebener Besucherzahl.

```
;; berechnet den Profit eine Filmvorführung aus 2000 WE Fixkosten
;; und 50 WE Kosten pro Besucher bei gegebener Besucherzahl und
;; 500 WE Kartenpreis
(define profit
  (lambda [besucherzahl]
    (- (* 500 besucherzahl)
      (+ (* 50 besucherzahl) 2000))))
;; Beispielanwendungen
(= (profit 100) 43000)
```