



Hausaufgabe 3 – Programmieren mit Konstanten und Funktionen

Bearbeiten Sie dieses Hausaufgabenblatt alleine. Stellen Sie dabei sicher, dass Sie die gesetzten Lernziele erreichen.

- Technische Voraussetzung für die Bearbeitung der Aufgaben ist, dass Sie Racket 6.10 installiert haben, als Sprache BSL (Anfänger) ausgewählt ist und das Teach-pack `2htdp/image` am Anfang der Datei über

```
(require 2htdp/image)
```

importiert wird.

- Bitte beachten Sie auch die Vorlesungshomepage (<http://ps.informatik.uni-tuebingen.de/teaching/ws17/info1/>)

Viel Erfolg!

Lernziele

Nach der Bearbeitung dieses Arbeitsblattes haben Sie gelernt

- ✓ Konstanten zu definieren
- ✓ Magic Literals zu vermeiden
- ✓ Funktionen zu definieren
- ✓ bestimmte Arten der Wiederholung zu vermeiden

Nach dieser Übung sollen Sie in der Lage sein, selbstständig Funktionen zu definieren, um kleine Programme strukturieren und Ausdrücke einfacher wiederverwenden zu können.

```
(define CHESSBOARD
  (above
    (beside (rectangle 20 20 "solid" "white")
      (rectangle 20 20 "solid" "black")
      (rectangle 20 20 "solid" "white")
      (rectangle 20 20 "solid" "black"))
    (beside (rectangle 20 20 "solid" "black")
      (rectangle 20 20 "solid" "white")
      (rectangle 20 20 "solid" "black")
      (rectangle 20 20 "solid" "white"))
    (beside (rectangle 20 20 "solid" "white")
      (rectangle 20 20 "solid" "black")
      (rectangle 20 20 "solid" "white")
      (rectangle 20 20 "solid" "black"))
    (beside (rectangle 20 20 "solid" "black")
      (rectangle 20 20 "solid" "white")
      (rectangle 20 20 "solid" "black")
      (rectangle 20 20 "solid" "white")))))

(define EXAMPLE-GAME (place-image/align
  (circle 10 "solid" "red")
  40 0 "left" "top"
  (place-image/align
    (circle 10 "solid" "red")
    20 40 "left" "top"
    (place-image/align
      (circle 10 "solid" "blue")
      60 60 "left" "top"
      CHESSBOARD))))
```

Aufgabe 1: Code verbessern und Wiederholungen vermeiden

Sie arbeiten in einer Firma und sollen ein Spiel für zwei Spieler entwickeln. Bisher wurde Ihnen vom Management nur weitergegeben, dass die Spielfläche sehr ähnlich wie bei Schach ist, aber 4 x 4 Flächen hat. Die Spielfiguren sollen rote und blaue Kreise sein. Ein Kollege von Ihnen fängt direkt an darauf los zu programmieren und zeigt Ihnen stolz den obigen Code.

Sie erkennen direkt, dass sich einige Stellen im Code häufiger wiederholen. Schreiben Sie den Code um und versuchen Sie dabei auf folgendes zu achten:

- Identifizieren Sie Magic Literals und ersetzen Sie diese durch geeignete Konstantendefinitionen
- Identifizieren Sie andere wörtliche Wiederholungen und ersetzen Sie auch diese durch geeignete Konstanten
- Verwenden Sie Funktionsdefinitionen, um Wiederholung von Ausdrücken zu vermeiden, die sich in Variationspunkten unterscheiden.

Die Manager aus dem Stockwerk über Ihrem Büro kommen nun zu Ihnen und fordern, dass alle schwarzen Flächen dunkelgrau dargestellt werden.

- Beschreiben Sie kurz in einem Kommentar, wie die Umsetzung dieser Anforderung in Ihrer Version des Programms verglichen mit der Version Ihres Kollegen ausfallen würde. Wie hoch schätzen Sie den jeweiligen Aufwand ein?

Bewertungsschema (Gesamtpunktzahl 60 pt)

- ☐ 5 pt Es wurden Magic Literals durch Konstanten ersetzt
- ☐ 5 pt Es wurden geeignete Namen für die Konstanten gewählt
- ☐ 15 pt Alle wörtlichen Wiederholungen wurden durch geeignete Konstanten ersetzt
- ☐ 15 pt Es wurden Funktionsdefinitionen sinnvoll zur Vermeidung von nicht-wörtlichen Wiederholungen verwendet
- ☐ 10 pt Lesbare Formatierung mit Zeilenumbrüchen, Einrückung, ...
- ☐ 5 pt Es handelt sich um ein Refactoring von CHESSBOARD (d.h. das Verhalten von CHESSBOARD hat sich nicht geändert).
- ☐ 5 pt Es handelt sich um ein Refactoring von EXAMPLE-GAME (d.h. das Verhalten von EXAMPLE-GAME hat sich nicht geändert).

Aufgabe 2: Eigene Funktionen schreiben

Programmieren Sie eine Funktion, die drei Parameter akzeptiert und ein Bild erzeugt. Die ersten zwei Parameter sind Farben, die im Bild vorkommen sollen. Der dritte Parameter ist eine Zahl zwischen 1 und 100, die Sie im Bild irgendwie verwenden sollen. Definieren Sie die Funktion mit dem Namen `generate-image`:

```
(define (generate-image color1 color2 number)
  ...)
```

Definieren Sie weitere Funktionen und Konstanten, um ein übersichtliches Programm abzugeben.

Bewertungsschema (Gesamtpunktzahl 40 pt)

- ☐ 4 pt `generate-image` ist definiert und akzeptiert drei Parameter
- ☐ 4 pt `generate-image` erzeugt ein Bild
- ☐ 12 pt `generate-image` erzeugt unterschiedliche Bilder für unterschiedliche Argumente
- ☐ 8 pt Lesbare Formatierung mit Zeilenumbrüchen, Einrückung, ...
- ☐ 8 pt "Don't Repeat Yourself" wurde beachtet
- ☐ 4 pt Die erzeugten Bilder sind interessant, lustig, beeindruckend, schön oder anderweitig sehenswert