



Hausaufgabe 4 – Bedeutung der BSL und Programmieren mit Animate

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

Verwenden Sie wieder die Sprache BSL (How to Design Programs — Anfänger).

Viel Erfolg!

Lernziele

Nach der Bearbeitung dieses Arbeitsblattes haben Sie gelernt

- ✓ den Ausdruck in Auswertungsposition zu identifizieren
- ✓ die Auswertungsschritte eines BSL-Programmes vorherzusagen
- ✓ mit `animate` bewegte Bilder zu beschreiben.

Aufgabe 1: Struktur und Bedeutung der BSL

In dieser Aufgabe arbeiten Sie mit den folgenden Beispielprogrammen:

a.

```
(define (f x y)
  (cond
    [(< x y) (+ x y)]
    [(> x y) (* x y)]))

(* 4 (f (+ 3 9) (+ 4 6)))
```

b.

```
(define (simple-color? text)
  (or (string=? text "red")
      (string=? text "green")
      (string=? text "blue")))

(simple-color? "green")
```

Bewertungsschema (Gesamtpunktzahl 30 pt)

- ☐ ☐ 3 pt Es wurde mindestens ein richtiger Auswertungsschritt angegeben.
- ☐ ☐ 3 pt Alle angegebenen Auswertungsschritte sind richtig.
- ☐ ☐ 3 pt Die angegebenen Auswertungsschritte sind vollständig und führen zur richtigen Bedeutung.
- ☐ ☐ 3 pt Es wurde mindestens ein richtiger Auswertungskontext angegeben.
- ☐ ☐ 3 pt Alle Auswertungskontexte sind angegeben und richtig.

Bearbeiten Sie die folgenden Teilaufgaben für jedes der Beispielprogramme.

1. Was bedeuten die Beispielprogramme?
2. Listen Sie alle Auswertungsschritte bei der Auswertung der Beispielprogramme als einzelne Ausdrücke auf.
3. Markieren Sie jeweils den Ausdruck in Auswertungsposition.

Zur Markierung der Auswertungsposition nutzen Sie bitte folgende Funktionsdefinition:

```
(define (mark x) x)
```

Beispiel. Die Auswertung von `(+ 1 2)` ergibt:

```
(* [mark (+ 1 2)] 5)
;-->
[mark (* 3 5)]
;-->
15
```

Aufgabe 2: Programmieren mit Animate

Als Teil eines Computerspiels soll ein Ball auf dem Bildschirm nach links und rechts bewegt werden können. Solange die Graphikabteilung die Bilder für den Ball noch nicht fertiggestellt hat, experimentieren die Programmierer mit der Steuerung eines Kreises.

Bevor der Ball aber vom Spieler hin- und herbewegt werden kann, soll zunächst ein Prototyp erstellt werden, in dem der Ball sich von alleine bewegt. Hier die Definition eines Balles und einer leeren Szene:

```
; Image of a red ball
(define BALL (circle 20 "solid" "red"))

; Empty scene to place balls in
(define SCENE (empty-scene 200 80))
```

1. Definieren Sie eine Funktion `place-ball`, die eine `x` und eine `y` Koordinate als Eingaben erwartet und ein Bild berechnet, bei dem sich der Ball an der gegebenen Position befindet.

Nun verwenden wir die Funktion `(animate FUN)` aus der `2htdp/universe` Bibliothek. Weitere Informationen zu `animate` finden Sie im Skript, Kapitel 2.

2. Programmieren Sie eine Funktion `create-ball-scene`, die einen Zählerwert als Argument nimmt und dann `place-ball` aufruft und den Zählerwert als `x`-Koordinate übernimmt. Die `y`-Koordinate soll dabei konstant bleiben.
3. Rufen Sie `(animate create-ball-scene)` auf und beobachten Sie das Ergebnis.
4. Verbessern Sie Ihre Animation. Verändern Sie die Definition von `create-ball-scene`, so dass der Ball nicht den Rand der Szene schneidet. Stoppen Sie die Bewegung des Balles, sobald dieser am rechten Rand angekommen ist.
5. Erweitern Sie Ihre Animation. Nachdem der Ball auf der rechten Seite angekommen ist, soll er sich zur Ausgangsposition zurückbewegen.
6. Letzte Änderung Ihrer Animation. In Ausgangsposition zurückgekehrt soll sich der Ball nun bis zur Mitte der Szene bewegen und dort stehen bleiben. Die Animation des Balles soll damit insgesamt ergeben: links-nach-rechts, rechts-nach-links, links-zur-mitte.
7. Führen Sie ein "Refactoring" durch. Verschieben Sie magische Literale in Konstantendefinitionen, lagern Sie Wiederholungen in Hilfsfunktionen aus und vereinfachen Sie konditionale Ausdrücke. Achten Sie darauf, dass sich das Verhalten Ihres Programmes dabei nicht verändert. Beschreiben Sie in einem kurzen Kommentar Ihre Konstantendefinitionen und Hilfsfunktionen.

Bewertungsschema (Gesamtpunktzahl 70 pt)

- ☐ 10 pt Die Funktion `place-ball` ist syntaktisch richtig definiert (4 pt), nimmt die richtigen Argumente (2 pt) und ergibt eine Szene (2 pt) mit positioniertem Ball (2 pt).
- ☐ 10 pt Die Funktion `create-ball-scene` ist syntaktisch richtig definiert (4 pt), nimmt ein Argument (2 pt) und ergibt eine Szene (2 pt), wobei das Argument zur Positionierung des Balles verwendet wird (2 pt).
- ☐ 10 pt Der Ball bewegt sich am Anfang von links nach rechts.
- ☐ 10 pt Der Ball bewegt sich am Anfang von links nach rechts und danach von rechts nach links.
- ☐ 10 pt Der Ball bewegt sich am Anfang von links nach rechts, danach von rechts nach links und schließlich in die Mitte und stoppt dort.
- ☐ 10 pt Der Ball bewegt sich bis zum Rand, schneidet diesen aber niemals.
- ☐ 10 pt Sinnvolle Konstanten und Hilfsfunktionen sind definiert und kommentiert.

Hinweis zur Abgabe auf dem Handin-Server

Achten Sie bitte bei der Abgabe Ihrer Hausaufgabe darauf, dass Sie alle Aufrufe von `animate` auskommentiert haben. D.h. zur Abgabe, ändern Sie

```
(animate create-ball-scene)
```

zu

```
; (animate create-ball-scene)
```