

**Übungen zur Vorlesung**  
**Deklarative Programmierung: Sommersemester 2018**

Nr. 1, Abgabe bis 23.4.2018 11:59 Uhr

**Hinweis:** Dieses Übungsblatt soll dazu dienen, die Arbeitsumgebung einzurichten und sich mit den Werkzeugen sowie ILIAS vertraut zu machen. Deshalb soll dieses Übungsblatt ausnahmsweise individuell bearbeitet und auch von jedem Gruppenmitglied einzeln über ILIAS abgegeben werden!

**Aufgabe 1.1:** Don't call us, we call you!

0 Punkte

In dieser Aufgabe sollen Sie dafür sorgen, dass wir Sie über die ILIAS-Plattform erreichen können. Alle wichtigen Mitteilungen zur Vorlesung und zur Übung werden über ILIAS versendet, weshalb Sie dafür sorgen sollen, dass Sie diese Nachrichten erreichen. Die über ILIAS versendeten Nachrichten werden an Ihren HRZ-Account gesendet. Bitte richten Sie über folgendes Formular eine Weiterleitung an eine andere E-Mail-Adresse ein, falls Sie den HRZ-Account nicht regelmäßig nutzen:

`https://admin.students.uni-marburg.de/change-forward.html`

**Aufgabe 1.2:** Einrichten der Arbeitsumgebung

4 Punkte

Zur Bearbeitung der Übungsaufgaben benötigen Sie die *DrRacket* - Entwicklungsumgebung. In dieser Aufgabe richten Sie diese Umgebung in vier einfachen Schritten ein:

- Laden Sie *DrRacket* (<http://racket-lang.org>) herunter und führen Sie die Installation durch.
- Stellen Sie als Sprache „How To Design Programs - Anfänger“ ein (**Sprache** → **Sprache auswählen**).
- Fügen Sie die Teachpacks universe und 2htdp/image hinzu (**Sprache** → **Teachpack hinzufügen**).
- Werten Sie den Ausdruck `(+ 17 25)` im *Interaktionsbereich* von *DrRacket* aus, fertigen Sie einen Screenshot an und geben Sie diesen ab.

**Aufgabe 1.3: Ausdrücke**

4 Punkte

Programmieren Sie einen Ausdruck, der

- a) die Summe der Zahlen 5, 7 und 9 berechnet
- b) die Zahl 42 in mindestens 3 Schritten berechnet
- c) „Computer sagt ...“ aus zwei Zeichenketten zusammensetzt. Ersetzen Sie dabei ... durch „Ja“ oder „Nein“.
- d) mindestens 3 Bilder miteinander kombiniert.

**Aufgabe 1.4: Reduktion & Äquivalenz**

4 Punkte

Reduzieren Sie die folgenden Ausdrücke schrittweise, bis Sie den zugehörigen Wert erhalten (vgl. Skript Kapitel 1.4):

- a) `(+ (* 4 (- 15 5)) (/ 12 6))`
- b) `(* (string->number "4") (- 33 (string-length "Spoilers")))`

Zeigen oder widerlegen Sie, dass die folgenden Ausdrücke äquivalent sind:

- c) `(* (* 4 5) (string->number "5"))`  $\equiv$   
`(* (+ 3 7) (+ (string->number "5") 5))`
- d) `(+ (string-length "Programmiersprachen") 5)`  $\equiv$   
`(+ 12 (string-length (replicate 5 "hi")))`

**Aufgabe 1.5: Eigene Funktion**

4 Punkte

Programmieren Sie eine Funktion:

- a) `(sqsum a b)`, die die Quadratsumme der beiden übergebenen Parameter berechnet (Bsp: `(sqsum 2 3) = 13`).
- b) `(center img)`, die das übergebene Bild zentriert in einer leeren Szene darstellt (vgl. Kapitel 1.2).