

Check und Prepare

Vorbereitende Übungen



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Autoren: Nhan Huynh und Darya Nikitina
Fachbereich: Informatik
Übungsblatt: 05

Version: 20. November 2021
Semesterübergreifend

V1 DrRacket installieren

Installieren Sie zu Beginn DrRacket von folgendem Link auf Ihrem Rechner:

<https://racket-lang.org/download/>

In Abbildung 1 sehen Sie, wie die Oberfläche dann aussieht. Vergewissern Sie sich, dass links unten (im Screenshot gelb hinterlegt) auch *Advanced Student* steht. Andernfalls klicken Sie auf die im Bild gelb markierte Fläche und stellen es darauf um.

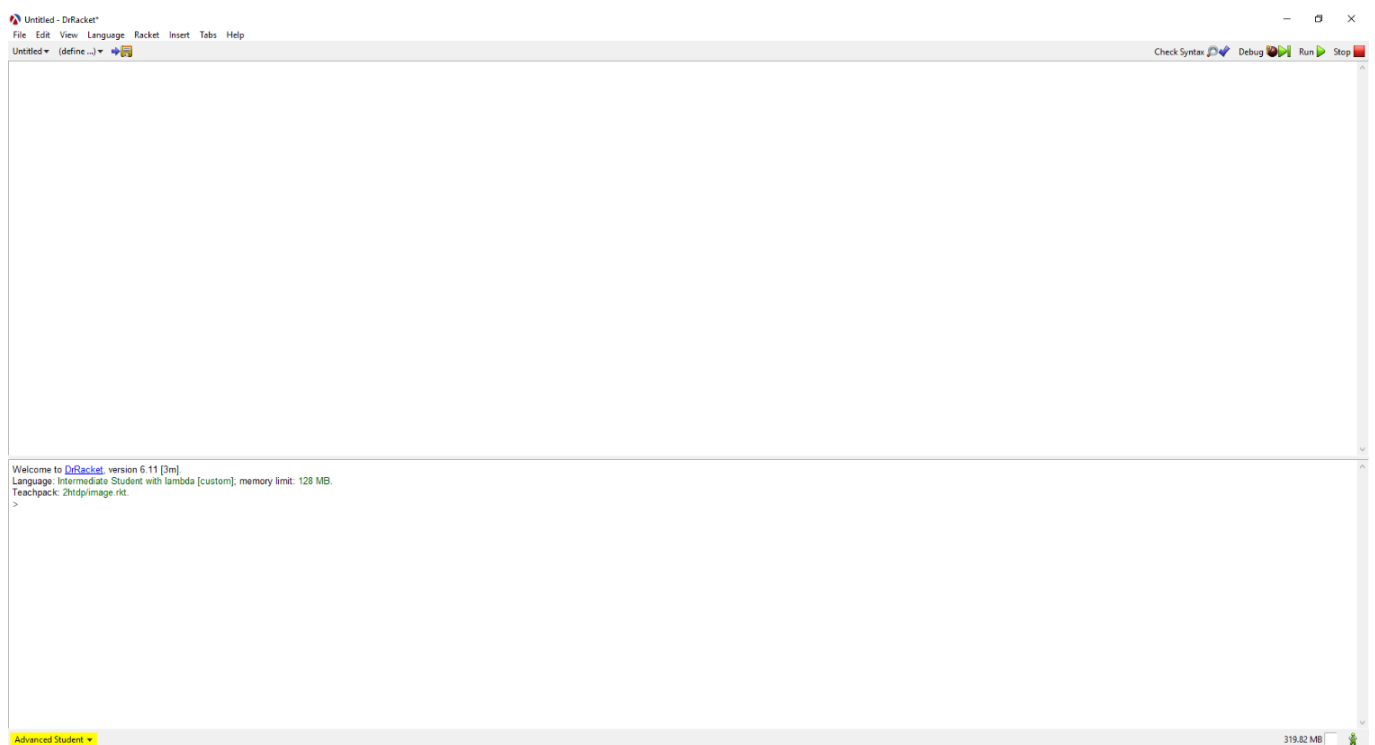


Abbildung 1: Oberfläche von DrRacket

Wichtig: Denken Sie an den Vertrag und die drei Tests bei jeder Funktion.

Achtung: Beachten Sie, dass ein Funktionsaufruf erwartet wird, falls Sie eine öffnende Klammer schreiben. Wenn Sie das nicht beachten, erhalten Sie folgende Fehlermeldung:

function call: expected a function after the open parenthesis, but received XYZ

V2 Temperaturumrechnung



Im angloamerikanischen Maßsystem wird die Temperatur nicht wie hierzulande in Grad Celsius gemessen, sondern in Grad Fahrenheit. Da Sie damit nichts anfangen können, wollen Sie sich nun eine Funktion `fahr->cel` definieren, welche die aktuelle Temperatur in Grad Fahrenheit übergeben bekommt und in Grad Celsius umwandelt.

Hinweis: Für eine gegebene Temperatur T_F in Grad Fahrenheit berechnet sich die dazugehörige Temperatur T_C in Grad Celsius über den folgenden Zusammenhang:

$$T_C = (T_F - 32) \cdot \frac{5}{9}$$

V3 Volumen eines Tetraeders



Das Tetraeder ist ein Körper mit vier dreieckigen Seitenflächen. Sein Volumen berechnet sich über die Formel $V(a) = \frac{\sqrt{2}}{12} a^3$, wobei a hier die Länge einer Kante ist. Sie sollen in dieser Aufgabe nun eine Funktion `tetrahedron-volume` schreiben, die für eine übergebene Kantenlänge a das Volumen des dazugehörigen Tetraeders zurückgibt. Gehen Sie dazu in folgenden Schritten vor:

1. Definieren Sie eine Funktion `pow3`, welche einen Parameter x bekommt, und den Wert x^3 zurückgibt. Erstellen Sie außerdem eine Konstante k , mit dem Wert $\frac{\sqrt{2}}{12}$.
2. Nutzen Sie die zwei vorherigen Schritte, um nun die Funktion `tetrahedron-volume` zu definieren, welche nur einen Parameter a bekommt.

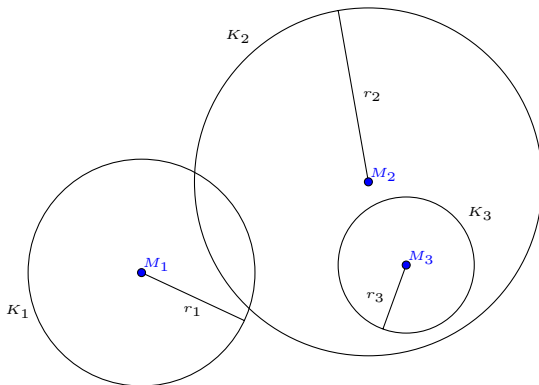
V4 Relative Lage zweier Kreise zueinander



Wir wollen die Lage zweier Kreise zueinander bestimmen. Definieren Sie dazu eine Prozedur `circles-position`, welche die Zahlen $x_1, y_1, r_1, x_2, y_2, r_2$ in dieser Reihenfolge entgegennimmt und einen String zurückgibt, welche die Lage der beiden Kreise zueinander beschreibt. Dabei sind x und y die Koordinaten eines Kreismittelpunktes und r der Radius des Kreises.

Zurückgegeben werden soll **"Intersect"** bei einem Schnitt der beiden Kreise, **"External"** bei keinerlei Überlappung oder **"Interior"** wenn einer der beiden Kreise vollständig im anderen liegt. Eine Berührung der beiden Kreise, egal ob von innen oder von außen, soll als Schnitt der beiden Kreise erkannt werden.

Zum besseren Verständnis finden Sie im Anschluss an die Aufgabe ein Beispiel und die mathematische Konkretisierung des Sachverhalts (dabei steht d für den Abstand der Mittelpunkte). Bei den Kreisen K_1 und K_2 im Beispiel sollte **"Intersect"**, bei den Kreisen K_1 und K_3 sollte **"External"** und bei den Kreisen K_2 und K_3 sollte **"Interior"** zurückgegeben werden.



$$\text{circles-position} = \begin{cases} \text{Interior} & \text{if } d < |r_1 - r_2| \\ \text{External} & \text{if } r_1 + r_2 < d \\ \text{Intersect} & \text{otherwise} \end{cases}$$

Zur Berechnung von d verwenden Sie den euklidischen Abstand der Mittelpunkte der zwei zu vergleichenden Kreise. Der Abstand zweier Punkt $p_1 = (x_1, y_1)$ und $p_2 = (x_2, y_2)$ ist folgendermaßen definiert $d_{p_1, p_2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

V5 Euklidischer Algorithmus



In der folgenden Aufgabe soll das Konzept der Rekursion verinnerlicht werden. Dazu werfen wir einen Blick auf eine Version des euklidischen Algorithmus, welcher Ihnen vielleicht schon aus der Mathematik bekannt ist. Für zwei natürliche Zahlen a und b lässt sich mit ihm der größte gemeinsame Teiler der beiden Zahlen berechnen. Dabei geht der Algorithmus wie folgt vor:

Gilt $b = 0$ so wird a zurückgegeben, ist hingegen $a = 0$ so wird b zurückgegeben. Gilt $a > b$ so wird der Algorithmus mit einem neuen $\hat{a} = a - b$ und dem "alten" b aufgerufen. Im anderen Fall wird der Algorithmus mit dem "alten" a und einem neuen $\hat{b} = b - a$ aufgerufen. Definieren Sie eine Prozedur `(euclid a b)`, welche diese Version des euklidischen Algorithmus rekursiv umsetzt.

V6 Listenausdrücke auswerten



Teil A: Werden die folgenden Ausdrücke ohne Fehler durch Racket ausgeführt? Falls nein, begründen Sie wo und warum es zu Problemen kommt.

1. `(cons 1 (cons 2 (cons 3)))`
2. `(cons 1 (list 2 (list (list 3 + 4))))`
3. `(list (cons empty 1)(cons 2 empty)(cons 3 empty))`
4. `(first empty)`

Teil B: Liefern die folgenden Listenausdrücke dasselbe Ergebnis zurück?

1. `(cons 1 (cons 2 (cons 3 empty)))` und `(list 1 2 3 empty)`
2. `(cons (list "(list)") empty)` und `(list "list" empty)`
3. `(list 7 "*" 6 "=" 42)` und `(cons 7 (cons "*" (cons 6 (cons "=" (list 42)))))`

Teil C: Gehen Sie nun von folgendem Codeschnipsel aus:

```
1 (define A (list (cons 1 empty)(list 7) 9 ))
2 (define B (cons 42 (list "Hello" "world" "!")))
```

Was liefern die folgenden Aufrufe zurück?

1. `(first (rest A))`
2. `(res (first A))`
3. `(append (first B)(rest (rest A))(first A))`

V7 Strukturausdrücke auswerten



Gegeben sei folgende Strukturdefinition:

```
1 (define-struct my-pair (one two))
```

Was liefern die folgenden Aufrufe zurück?

1. `(my-pair? (make-my-pair "a" "b"))`
2. `(make-my-pair 1 (make-my-pair 2 empty))`
3. `(* (my-pair-two (make-my-pair 1 2))(my-pair-one (make-my-pair 3 4)))`

V8 Listen in Strukturen



Gegeben ist ein Struct-Typ `abc` mit zwei Feldern `a` und `b`. Definieren Sie eine Funktion `foo` mit einem Parameter `p`. Falls `p` vom Typ `abc` und zudem der Wert im Feld `b` von `p` eine Liste ist, liefert `foo` eine Liste zurück, deren erstes Element der Wert von Feld `a` in `p` ist, und der Rest der zurückgelieferten Liste ist die Liste im Feld `b` von `p` (also eine Liste in der Liste). Andernfalls liefert `foo` einfach `false` zurück.

V9 Suche in Zahlenliste



Definieren Sie eine Funktion `contains-x?`. Diese bekommt eine Liste und eine Zahl übergeben und liefert genau dann `true` zurück, wenn die übergebene Zahl mindestens einmal in der übergebenen Liste vorkommt.

V10 Duplikate in Zahlenliste



Definieren Sie eine Prozedur `duplicates?` mit einem Parameter `lst`, die genau dann `true` zurückliefert, wenn eine Zahl mehr als einmal in `lst` vorkommt.

Hinweis: Können Sie hier vielleicht Ihre Funktion aus Aufgabe V9 verwenden?

V11 Verschachtelte Listen



Definieren Sie eine Prozedur `duplicates-deep?` mit einem Parameter `deep-lst`. Es wird erwartet, dass `deep-lst` entweder eine Zahl oder eine Liste ist, deren Elemente wiederum Zahlen oder Listen sind usw. (Liste von Listen von Zahlen). Die Prozedur `duplicates-deep?` soll `true` oder `false` zurückliefern, und zwar `true` genau dann, wenn mindestens eine Zahl mehr als einmal vorkommt.

Hinweis: Schreiben Sie sich eine Hilfsfunktion `collect` mit zwei Parametern `lst` und `oracle`. Nutzen Sie `oracle` als Akkumulator in dem Sie alle bereits vorgekommenen Zahlen in `oracle` speichern. Machen Sie also aus der verschachtelten Liste wieder eine normale Liste mithilfe von `collect`. Die Funktion aus V10 kann Ihnen hier sehr hilfreich sein.

V12 Arithmetisches Mittel



Gegeben seien folgende Strukturdefinition:

```
1 (define-struct person (age sex))
2 (define-struct student (person id))
```

Eine Person hat ein Alter (als Zahl) und ein Geschlecht (als String). Ein Student wiederum besteht aus einer Person (vorheriges Struct) und einer Matrikelnummer (ein String).

Definieren Sie nun eine Funktion `mean-of-ages`. Diese bekommt eine Liste von Studenten übergeben und gibt das arithmetische Mittel ihrer Alter zurück.

Hinweis: Das arithmetische Mittel \bar{x} berechnen Sie für n Alter x_1, \dots, x_n mithilfe von:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{x_1 + \dots + x_n}{n}$$