



Grundlagen der Informatik 1

Sommersemester 2011

Dr. Guido Röbling
<https://moodle.informatik.tu-darmstadt.de/>

Übung 1 Version: 1.1

18. 04. 2011

1 Mini-Quiz

Kreuzen Sie bei den folgenden Aufgaben die richtige Lösung oder die richtigen Lösungen an.

1. Betrachten Sie die folgende Racket-Funktion:

```
1 (define (sth x)
2   (if (< x 5) 'x+1 'x*2))
```

Was wird hier definiert?

- Eine Prozedur. Ein Symbol. Eine Zahl.

Welches Ergebnis liefert der Aufruf von (sth 4) zurück?

- 5 8 Ein Symbol. Einen String.

2. Was bedeutet „Prozedurale Abstraktion“?

- Die Zerlegung eines großen Problems in kleinere Teilprobleme.
 Die Auslagerung von Berechnungen in eigene Prozeduren.
 Abstraktion dient dazu, Details einer Berechnung zu zeigen.

2 Fragen

1. Was sind die Strukturierungsmechanismen einer Programmiersprache? Nennen Sie für jeden Mechanismus in Racket ein Beispiel.
2. Nennen Sie die generellen Schritte beim Design von Programmen.
3. Wozu werden Symbole in Racket verwendet? Warum werden nicht stattdessen Strings verwendet?

3 Racket-Syntax (K)

3.1 Klammerung

Ergänzen Sie folgende Ausdrücke durch Klammern, so dass jeweils ein gültiger Racket-Ausdruck entsteht. Werten Sie anschließend diese Ausdrücke aus.

1. + + + 7 3 4 * 2 3

2. * + 1 9 - 4 + 1 1

3. not or true and true false or true false

3.2 Präfix-Notation

Übersetzen Sie die folgenden mathematischen Ausdrücke in äquivalente Racket-Ausdrücke. Die Ausdrücke sollen nicht vereinfacht (gekürzt oder berechnet) werden. 4^3 dürfen Sie durch eine entsprechende Multiplikation ersetzen.

1. $\frac{13-3}{6*3} * (100 - 32)$
2. $\frac{7*2}{28} * \frac{2}{3}$
3. $\frac{4^3}{8} + (4 - 12)$

4 Tetraedervolumen (K)

In der folgenden Aufgabe soll schrittweise eine Prozedur `volume` in Racket erstellt werden, die das Volumen eines Tetraeders mit gegebener Kantenlänge berechnet. Dabei soll nach dem Top-Down Ansatz vorgegangen werden.

Hinweis: Das Volumen V eines Tetraeders mit Kantenlänge a berechnet sich als $V(a) = \frac{\sqrt{2}}{12}a^3$. Die Funktion zur Berechnung der Quadratwurzel lautet in Racket `sqrt`.

1. Geben Sie den Vertrag, die Beschreibung und ein Beispiel für die Prozedur `volume` an. `volume` soll einen Wert für die Kantenlänge konsumieren und das zugehörige Tetraedervolumen berechnen.
2. Schreiben Sie nun die Definition der Prozedur in Racket auf. Gehen Sie hierbei davon aus, dass Sie auf Ihrer Wunschliste eine weitere Prozedur `pow3: number -> number` (berechnet x^3 für eine Eingabe x) sowie eine Konstante `k` (für $\frac{\sqrt{2}}{12}$) haben. Geben Sie außerdem einen Test für die Prozedur `volume` an.
3. Erstellen Sie nun die Prozedur `pow3` mit Vertrag, Beschreibung und Beispiel, und definieren Sie die Konstante `k` mit einem Wert von $\frac{\sqrt{2}}{12}$.
4. Überlegen Sie sich, wie Sie bei der Programmierung von `volume` nach dem Bottom-Up Verfahren vorgegangen wären.

5 Auswertung von Ausdrücken (K)

Gehen Sie von folgender Racket-Prozedur aus:

```

1 ;; weighted-average : number number number -> number
2 ;; calculates a weighted average of 3 given numbers x,y,z
3 ;; using the formula (x+y+y+z)/4
4 ;; Example: (weighted-average 3 6 7) should produce 5.5
5
6 ;; Definition:
7 (define (weighted-average x y z)
8   (/ (+ x y y z) 4))
9
10 ;; Test
11 (check-expect (weighted-average 3 8 1) 5)
12 (check-within (weighted-average 3 1 4) 2.25 0.01)

```

Überlegen Sie, wie folgende Ausdrücke in applikativer und normaler Auswertungsreihenfolge ausgewertet werden, und schreiben Sie zu jedem Ausdruck eine Variante auf (siehe Folie T1.58ff).

1. `(weighted-average (weighted-average 15 10 5)/ 36 9)(* 2 3)`
2. `(weighted-average (* 3 4)(weighted-average 4 13 2)(weighted-average 31 50 13))`

6 Steuern

Die Steuern in Land X ergeben sich aus dem Einkommen multipliziert mit dem Steuersatz. Der Steuersatz ist wiederum 0,5% je tausend Euro Einkommen (abgerundet). So würde ein Einkommen von 40.000 Euro einem Steuersatz von $0,5\% * 40 = 20\%$ und einer zu zahlenden Steuer von 8.000 Euro entsprechen, ein Einkommen von 23.700 Euro einem Steuersatz von 11,5% etc.

Hinweis: Zum Runden von Werten kann die Funktion `round` verwendet werden. Aber Vorsicht: diese Funktion rundet ab 0.5 auf, siehe auch <http://en.wikipedia.org/wiki/Rounding!>

1. Schreiben Sie eine Funktion `get-taxrate`, die ein Einkommen konsumiert und den Steuersatz (in Prozent, d.h. 40 für 40%) produziert. Schreiben Sie dazu eine Funktion `round-down`, die eine gegebene Zahl abrundet. Geben Sie zu beiden Funktionen jeweils Vertrag, Beschreibung, Beispiel und mindestens zwei Tests an.
2. Schreiben Sie eine Funktion `get-income`, die ein Einkommen konsumiert und das Einkommen nach Abzug der Steuer produziert. Geben Sie auch hierzu Vertrag, Beschreibung, Beispiel und mindestens zwei Tests an.

Hausübung

Die Vorlagen für die Bearbeitung werden im Lernportal Informatik bereitgestellt. Kommentieren Sie Ihren selbst erstellten Code. Die Hausübung muss bis zum Abgabedatum im Lernportal Informatik abgegeben werden.

Der Fachbereich Informatik misst der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch die strikte Verfolgung von Plagiarismus. Mit der Abgabe Ihrer Hausübung bestätigen Sie, dass Sie bzw. Ihre Gruppe alleiniger Autor des gesamten Materials sind. Falls Ihnen die Verwendung von Fremdmaterial gestattet war, so müssen Sie dessen Quellen deutlich zitieren.

Falls Sie die Hausübung in einer Lerngruppe bearbeitet haben, geben Sie dies bitte deutlich bei der Abgabe an. Alle anderen Mitglieder der Lerngruppe müssen als Abgabe einen Verweis auf die gemeinsame Bearbeitung einreichen, damit die Abgabe im Lernportal Informatik auch für sie bewertet werden kann. Beachten Sie dazu die Hinweise bei der Aufgabenabgabe im Lernportal Informatik!

Abgabedatum: Freitag, 29. 04. 2011, 18:00 Uhr

Denken Sie bitte daran, Ihren Code hinreichend gemäß den Vorgaben zu kommentieren (Racket: Vertrag, Beschreibung und Beispiel sowie zwei Testfälle pro Funktion; Vertrag, Beschreibung und Beispiel für jede `local` definierte Funktion; Vertrag (ohne Namen) und kurze Beschreibung für jeden `lambda`-Ausdruck; Java: JavaDoc). Zerlegen Sie Ihren Code sinnvoll und versuchen Sie, wo es möglich ist, bestehende Funktionen wiederzuverwenden. Wählen Sie sinnvolle Namen für Hilfsfunktionen und Parameter.

7 Aller Anfang...

Installieren Sie DrRacket von <http://racket-lang.org/> auf Ihrem Rechner oder rufen Sie DrRacket mit dem Befehl 'racket' auf den Rechnern der RBG auf. Machen Sie sich mit den Grundfunktionen des Programms vertraut. Einen guten Ausgangspunkt bietet die kurze Tour auf <http://docs.racket-lang.org/quick/index.html>. Probieren Sie einige der Beispiele aus der Vorlesung aus, z.B. T1.19f., T1.33f, T2.5f., T2.14f. Sie finden diese auch im Portal in der *Datenbank*. **Wichtig:** Stellen Sie das Sprachlevel für diese und die folgenden Aufgaben auf *How To Design Programs -> Anfänger* (bzw. *Beginning Student* in der englischen Version).

Hinweis für die folgenden Aufgaben: Kommentieren Sie **alle** von Ihnen definierten Funktionen mit Vertrag, Beschreibung, Beispiel sowie mindestens zwei Tests. Dies ist ein Bestandteil der Bewertung! Versuchen Sie, Teilprobleme sinnvoll in Hilfsfunktionen auszulagern. Außerdem dürfen Sie Funktionen und Hilfsfunktionen aus anderen Aufgabenteilen wiederverwenden.

8 Hallo Lernportal Informatik! (1 Punkt)

Die Lösung Ihrer Hausübungen müssen Sie online im Lernportal Informatik einreichen. Besuchen Sie daher mit ihrem Browser die URL <https://moodle.informatik.tu-darmstadt.de/>.

1. Als erstes sollten Sie gegebenenfalls die eingestellte Sprache anpassen, z.B. auf *Deutsch - Du (de_du)* oder *English*. Die Auswahlmöglichkeiten finden Sie oben rechts auf der Startseite des Portals.
2. Klicken Sie auf den Kurs *Grundlagen der Informatik I, Sommersemester 2011*. Falls Sie bereits im Portal registriert sind, geht es weiter bei Schritt 5.
3. Legen Sie ein neues Benutzerkonto an, indem Sie auf den Button „Neuen Zugang anlegen?“ klicken. Füllen Sie das nun erscheinende Formular vollständig aus und achten Sie darauf, eine von Ihnen regelmäßig gelesene Mailadresse zu nutzen. Bitte klicken Sie am Ende auf „Meinen neuen Zugang anlegen (Registrierung)“. Korrigieren Sie ggf. Ihre Einträge, bis das Formular erfolgreich abgesendet werden konnte.
4. An die angegebene Emailadresse wird nun eine Bestätigung geschickt. Bitte öffnen Sie diese Mail und klicken Sie auf den darin enthaltenen Link; es öffnet sich wiederum eine Portalseite. Wechseln Sie nun zur Kursseite.
5. Überprüfen Sie Ihr Nutzerprofil, indem Sie (ganz oben) auf Ihren Namen klicken und anschließend auf den Karteireiter (auch Tab genannt) „Profil bearbeiten“: ist der Name vollständig und korrekt angegeben? Haben Sie Ihre Matrikelnummer korrekt angegeben? Ohne diese Angaben können wir für Sie am Ende keine Studienleistung in TUCaN melden!

Tipp: Klicken Sie auf die Schaltfläche *Zusätzliche Felder anzeigen*. Besonders wichtig ist der Eintrag „Forenbeiträge zusammenfassen“. Normalerweise sendet das Lernportal Informatik Ihnen bei jedem in einem von Ihnen gelesenen Forum neu geschriebenen Beitrag eine Email. Im Lauf des Semesters können das *sehr viele* Emails pro Tag werden! Unter dem Eintrag *Forenbeiträge zusammenfassen* können Sie einstellen, dass Sie statt einer Email pro neuem Forenbeitrag nur einen täglichen „Digest“ erhalten wollen. Wir raten Ihnen *sehr* dazu, hier die Option „Ja, täglich eine E-Mail mit den Betreffs aller Forenbeiträge senden“ zu wählen, damit Ihr Emailfach im Lauf des Semesters nicht überläuft!

Hinweis: Die Matrikelnummer ist für andere Studierende nicht sichtbar.

6. Neben Foren und den Vorlesungsmaterialien finden Sie im Portal auch Aufgaben. Über diese geben Sie auch Ihre Hausaufgaben ab. Hausübungsabgaben können nur Sie, der Dozent und die Tutoren einsehen. Ihr Tutor wird versuchen, Ihre Lösung innerhalb von 7 Tagen nach der Abgabe zu korrigieren. Die Korrektur können Sie dann als Kommentar zu Ihrer Lösung lesen. Je früher Sie ihre Lösung abgeben, desto früher haben Sie in der Regel auch die Korrektur durch den Tutor!

Zusätzlich stehen im Portal die folgenden Hilfsmittel als separate Kurse für Sie bereit:

- Kurs „Grundlagen der Informatik I (semesterübergreifend)“: in diesem Kurs finden Sie die Vorlesungsunterlagen („Folien“), Vorlesungsaufzeichnungen, Wissenstests, Links zu Materialien und andere Elemente, die nicht an das konkrete laufende Semester gebunden sind (Faustformel: fast alles außer Ankündigungen und die Unterlagen zum Übungsbetrieb). Tragen Sie sich **unbedingt** in diesen Kurs ein!
- Kurs „Hilfen zur Nutzung des Portals“: hier finden Sie Information zum Lernportal Informatik und wie man es möglichst effizient nutzen kann. Das Lernportal Informatik kann sehr viel, und nicht alles davon sieht man auf den ersten Blick. Daher empfehlen wir Ihnen **sehr**, sich auch in diesen Kurs einzutragen und die dortigen Materialien durchzuarbeiten!

- Kurs „Tipps zum effektiven Studieren“: hier erhalten Sie Tipps rund um das Studium, etwa zum Bilden von Lerngruppen oder der Vorbereitung auf Klausuren. Eine Einschreibung in diesen Kurs ist daher ebenfalls ratsam. Zusätzlich arbeiten wir hier noch an Beiträgen mit Tipps zur Programmierung.

Achtung: Um den Punkt für diese Aufgabe zu erhalten, müssen Sie sich im *Übungsgruppenforum* Ihrer Übungsgruppe kurz vorstellen. Sie sollten dabei Ihren Namen angeben und sich kurz beschreiben (Hobbys und Interessen), aber *bitte ohne Ihre Matrikelnummer anzugeben*.

Sollten Sie Probleme mit der Bedienung des Lernportals Informatik haben, sprechen Sie mit Ihren Kommilitonen und probieren Sie, selbstständig eine Lösung zu finden—dazu dient unter Anderem der oben genannte Kurs mit den „Hilfen zur Nutzung des Portals“!. Sollten Sie absolut nicht weiterkommen, schreiben Sie eine Anfrage im dazu am besten passenden Forum oder fragen Sie Ihren Tutor. Generell hilft es, wenn Sie Ihre Frage klar formulieren und aufschreiben. Nur so kann ein Tutor oder ein anderer Ansprechpartner Ihnen schnell helfen. Sollte Ihr Tutor Ihnen nicht weiterhelfen können, wenden Sie sich an die Veranstalter.

Hinweis: Bitte geben Sie diese Selbstbeschreibung **vor** der Einreichung der zweiten Hausaufgabe dieses Übungsblatts ab, damit der Tutor sie bei der Korrektur direkt mit berücksichtigen kann!

9 Währungsumrechnung (5 Punkte)

Für die Durchführung von Überweisungen im internationalen Geldverkehr muss oft zwischen Währungen umgerechnet werden. Ihre Aufgabe ist es nun, eine Reihe von Prozeduren zu schreiben, die eine Konvertierung zwischen den verschiedenen Währungen ermöglicht.

Beachten Sie für diese Aufgaben insbesondere die Folien T1.42 (**cond**) und T1.47f (Umgang mit Symbolen)! Entsprechen die übergebenen Symbole keiner bekannten Währung (siehe die Tabelle unten), soll ein Fehler ausgelöst werden. Die Hinweise hierzu sowie zum *Testen* des Codes finden Sie auf Folie T1.29.

9.1 Konvertierung einer beliebigen Währung in Euro (2 Punkte)

Implementieren Sie zunächst eine Prozedur `convert-x-to-euro: number symbol -> number`. Diese konvertiert einen Geldbetrag sowie eines der in der ersten Spalte der nachfolgenden Tabelle angegebenen Symbole für eine gültige Währung:

Symbol	Währung	€1 = ...
'euro	Euro (€)	€ 1,00
'dollar	US-Dollar (US-\$)	US\$ 1,422
'pound	Britische Pfund Sterling (£)	£ 0,881
'yen	Japanische Yen (¥)	¥ 119,557
'franken	Schweitzer Franken (Fr)	Fr 1,313

Hinweis: Die Umrechnungskurse stammen vom 4. 4. 2011 und *müssen exakt* wie oben angegeben verwendet werden, damit unsere Tests erfolgreich sind.

Als Ergebnis soll der in € konvertierte Geldbetrag geliefert werden. Zur Illustration ein paar Beispiele:

- (`convert-x-to-euro 22 'euro`) ergibt 22, da hier nicht umgerechnet werden muss.
- (`convert-x-to-euro 239.114 'yen`) ergibt 2: ¥239,114 entsprechen €2.
- (`convert-x-to-euro 65.65 'franken`) ergibt 50.

Vergessen Sie nicht, Ihre Prozedur mit *Vertrag, Beschreibung, Beispiel* und *mindestens zwei Tests*—mit anderen Werten als den oben angegebenen!—zu dokumentieren! Beachten Sie dabei, dass die Ergebnisse in der Regel *Gleitkommazahlen* sind, so dass Ihre Tests mit (**check-within...**) arbeiten sollten, nicht mit (**check-expect...**).

9.2 Konvertierung von Euro in eine andere Währung (1 Punkt)

Implementieren Sie nun eine Prozedur `convert-euro-to-x`: **number symbol** \rightarrow **number**. Diese konsumiert einen Geldbetrag in Euro sowie die Angabe der *gewünschten* Währung und produziert den Geldbetrag in der gewünschten Währung.

So soll etwa (`convert-euro-to-x 100 'dollar`) das Ergebnis 142.2 ergeben: €100 entsprechen US-\$142,20. Vergessen Sie auch hier nicht die Dokumentation wie oben beschrieben inklusive Tests!

9.3 Wechseln von Währungen (2 Punkte)

Schreiben Sie nun eine Prozedur `convert-x-to-y`: **number symbol symbol** \rightarrow **number**. Diese konsumiert einen Geldbetrag in der durch das erste Symbol angegebenen Währung und konvertiert sie in einen Geldbetrag in der zweiten Währung. Nutzen Sie die beiden vorherigen Prozeduren, um die Prozedur `convert-x-to-y` zu implementieren.

Beachten Sie dabei die folgenden Regeln Ihrer (etwas unflexiblen) Bank:

- Die Bank rechnet grundsätzlich *jeden* Betrag—unabhängig von der Währung—erst in die „Hauswährung“ Euro um, bevor sie diesen Betrag in die Zielwährung umrechnet.
- Für jeden Umrechnungsschritt zwischen *verschiedenen* Währungen fällt eine Gebühr an. Diese beträgt 1% des umzurechnenden Wertes.
- Nur das „Umrechnen“ von Euro in Euro ist kostenlos.
- Die Gebühr darf **nicht** in den beiden bereits implementierten Prozeduren erhoben werden, da diese auch für interne Transaktionen genutzt werden und daher exakt arbeiten müssen. Die Gebühr darf also nur in der „Endkundenmethode“ `convert-x-to-y` erhoben werden.

Beispielsweise ergibt (`convert-x-to-y 25 'dollar 'yen`) ≈ 2060.088 , denn aus US-\$ 25 werden zunächst einmal etwa €17,5808. Hiervon behält die Bank 1% ($\approx 0,18$), bevor sie die nun nur noch $\approx €17,40$ in $\approx ¥2080,89$ umrechnet und wiederum 1% ($\approx 20,808$) einstreicht. Das Ergebnis ist also etwa ¥2060,08.

Vergessen Sie auch hier nicht die Dokumentation Ihrer Methoden!

Hinweis: Überlegen Sie sich, wie sie geschickt eine mehrfache Berechnungen der Beträge vermeiden können. Für „Experten“: die Nutzung von Elementen wie `set!` oder `local` ist gemäß Sprachlevel *nicht erlaubt*.