

## Hinweise zur Abgabe

Bitte reichen Sie Ihre Abgaben bis zum 06.11.2008 um 11 Uhr ein. Abgaben in elektronischer Form schicken Sie **per Email** an **Ihren** Tutor. Abgaben in Papierform werfen Sie bitte in den **Briefkasten** Ihrer Übungsgruppe im Geb. 051 im Erdgeschoss. Bei jeder Aufgabe ist angegeben, ob Sie elektronisch oder auf Papier abgegeben werden muss.

Bei allen Aufgaben, die Sie per Mail abgeben, müssen Sie sich an die Namenskonventionen der Aufgaben halten. Dies gilt sowohl für die Dateinamen der Abgabe, als auch für Namen von Funktionen. Bitte geben Sie bei der elektronischen Abgabe nur eine Zip-Datei ab. Diese muss alle in den Aufgaben angegebenen `.scm` Dateien (DrScheme) enthalten. Alle Dateien müssen sich in der Zip-Datei in einem Ordner befinden. Der Name dieses Ordners muss Ihrem Loginnamen für den Rechnerpool des Instituts für Informatik entsprechen. Geben Sie unter keinen Umständen Worddokumente usw. ab!

Achten Sie bei der Papierabgabe darauf, dass jedes Blatt Papier Ihrer Abgabe Ihren Namen, Ihre Übungsgruppe, die Blattnummer und den Namen Ihres Tutors trägt. Falls Ihre Papierabgabe aus mehreren Seiten besteht, tackern Sie die Blätter.

Sie können DrScheme im Pool verwenden (starten mit `drscheme`). Achten Sie darauf, dass Sie jeweils das richtige Sprachlevel ausgewählt haben!

## Punktevergabe

Um für die Programmieraufgaben Punkte zu erhalten, folgen Sie den Konstruktionsanleitungen der Vorlesung.

### 1 Aufgabe

*[Sprache: Die Macht der Abstraktion - Anfänger, 7 Punkte]*

Nach Klicken auf **Syntaxprüfung** zeigt DrScheme, falls man mit der Maus über einen Variablennamen fährt, die Verwendungen bzw. die Definition an. Verwenden Sie dieses Feature, wenn Sie bei der lexikalischen Bindung Hilfe benötigen.

- a. Betrachten Sie das folgende Scheme-Programm.

```
(define x 2)           ; --> zwei
(define y -1)          ; --> minuseins
(define z -3)          ; --> minusdrei

(define f
  (lambda (x z)
    (+ (* x x) z y)
  ))

(f 4 -2)
```

Benennen Sie die Variablen `x`, `y` und `z`, die in den ersten drei Zeilen des Programms definiert werden, im kompletten Programm um. Der neue Name der Variable steht als Kommentar im Programm hinter dem Pfeil (`-->`). Achten Sie bei der Umbenennung auf die lexikalische Bindung. Benennen Sie keine Parameter der Funktion `f` um.

Berechnen Sie, nachdem Sie die Umbenennung durchgeführt haben, von Hand `(f 4 -2)`.

*Abgabe: Papier*

- b. Betrachten Sie das folgende Scheme-Programm.

```

(define x 1)
(define y 3)
(define z 5)

(define f
  (lambda (x)
    ((lambda (y)
      ((lambda (z)
        (+ z (* x y)))
         (+ x z)))
     (+ x y))))

(f y)

```

Benennen Sie hier alle lokalen Variablen, die innerhalb der Funktion `f` definiert werden, um. Verändern Sie nicht den Namen der Variablen `x`, `y` und `z` aus den ersten drei Zeilen des Programms.

Welches Ergebnis liefert der Ausdruck `(f y)`?

*Abgabe: Papier*

*Hinweis:* In diesen Aufgaben finden Sie keine Kommentare und Verträge zu den Funktionen. Hier können Sie an einem Beispiel sehen, dass es sehr wichtig ist, diese Informationen anderen Programmieren immer zur Verfügung zu stellen. Denn es kann auch bei kleinen Programmen schwer sein, die Funktionsweise der einzelnen Funktionen ohne Kommentare zu verstehen.

## 2 Aufgabe

[Sprache: Die Macht der Abstraktion - Anfänger, 3 Punkte]

Schreiben Sie eine Prozedur `card-type`, die den Umsatz einer Kreditkarte konsumiert und die eine entsprechende Kategorie als Zeichenkette zurückgibt. Verwenden Sie die Konstruktionsanleitung aus der Vorlesung: Schreiben Sie die Kurzbeschreibung auf, führen Sie eine Datenanalyse durch und schreiben Sie den Vertrag auf. Erstellen Sie dann das Gerüst und die Testfälle. Vervollständigen Sie danach den Rumpf der Prozedur und vergewissern Sie sich, dass die Tests erfolgreich laufen.

		Umsatz	<	15.000	⇒	Weiß
15.000	≤	Umsatz	<	50.000	⇒	Gold
50.000	≤	Umsatz	≤	150.000	⇒	Platin
150.000	<	Umsatz			⇒	Schwarz

*Abgabe:* elektronisch als `aufgabe4.scm`

## 3 Aufgabe

[Sprache: Die Macht der Abstraktion - Anfänger, 10 Punkte]

Vektoren im Raum:

- Definieren Sie einen neuen Record `3dvec` für Vektoren im  $\mathbb{R}^3$ . Orientieren Sie sich bei den Namen an der Vorlesung (04-data.pdf, S. 9). Erstellen Sie mindestens 5 Punkte als Datengrundlage.
- Schreiben Sie eine Funktion `3dvec-diff`, die für zwei Vektoren die Differenz berechnet. Erstellen Sie drei Testfälle. Verwenden Sie hierzu soweit möglich die Punkte aus dem ersten Aufgabenteil.
- Schreiben Sie eine Funktion `3dvec-length`, die für einen Vektor seine Länge berechnet. Erstellen Sie auch hier mindestens drei Testfälle.

*Abgabe:* elektronisch als `3dvec.scm`.