

Hinweise zur Abgabe

Bitte reichen Sie Ihre Abgaben bis zum 19.11.2009 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, (3+3) 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 a 3)           ; --> drei
(define b -10)        ; --> minuszehn
(define c 42)         ; --> antwort
(define f
  (lambda (a c)
    (+ a b c)))
(f c a)
```

Benennen Sie die Variablen `a`, `b` und `c`, 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.

(b) Betrachten Sie das folgende Scheme-Programm:

```
(define a 1)
(define b 3.5)
(define c 5)
(define g
  (lambda (a)
    ((lambda (b)
      ((lambda (c)
        (+ a b c))
        (+ a b c))))
    (+ a b c))))
(g c)
```

Benennen Sie hier alle lokalen Variablen, die innerhalb der Funktion `g` definiert werden, um, um. Verändern Sie nicht den Namen der Variablen `a`, `b` und `c` aus den ersten drei Zeilen.

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.

Abgabe: auf Papier

2 Aufgabe

[Sprache: Die Macht der Abstraktion, (2+2+2+4) Punkte]

- Eine Visakarte kann durch einen zusammengesetzten Datentypen mit den Eigenschaften `nummer` und `ablaufdatum` repräsentiert werden. Folgen Sie der Konstruktionsanleitung aus der Vorlesung und definieren Sie einen entsprechenden Record. Geben Sie auch einen Vertrag für den Konstruktor des Records an. Der Einfachheit halber sollen sowohl die Nummer als auch das Ablaufdatum der Karte als Zeichenketten (Strings) repräsentiert werden.
- Eine Masterkarte ist einer Visakarte sehr ähnlich, hat aber eine zusätzliche Eigenschaft `status`, welche einen der Werte `"silber"`, `"gold"` oder `"platin"` annehmen kann. Folgen Sie wieder der Konstruktionsanleitung aus der Vorlesung und definieren Sie einen entsprechenden Record. Geben Sie auch einen Vertrag für den Konstruktor des Records an.
- Eine Kreditkarte ist entweder eine Visakarte oder eine Masterkarte. Geben Sie eine Definitionen für diesen gemischten Datentypen an.
- Schreiben Sie eine Prozedur `VIP?`, die anhand einer gegebenen Kreditkarte feststellt, ob der Besitzer der Kreditkarte ein VIP ist. Ein VIP hat entweder eine Visakarte mit der Nummer `"9999 9999 9999 9999"` oder eine Masterkarte mit dem Status `"platin"`.

Abgabe: elektronisch als `kreditkarten.scm`

3 Aufgabe

[Sprache: Die Macht der Abstraktion, (3+3) Punkte]

Benutzen Sie den in der Vorlesung definierten Datentyp für Listen für folgende Prozeduren:

- Eine Prozedur `count-zeroes`, die die Anzahl von Nullen in einer Liste von Zahlen liefert.
- Eine endrekursive Variante `count-zeroes-it` von `count-zeroes`.

Abgabe: elektronisch als `list-procedures.scm`.

4 Aufgabe

[Sprache: Die Macht der Abstraktion, (4+4*) Punkte]

- Schreiben Sie eine Funktion `fib`, die die Fibonacci-Zahl einer Zahl `n` berechnet. Verwenden Sie hierzu direkt die Definition der Fibonacci-Zahlen:

$$\text{fib}(n) := \begin{cases} 0 & \text{falls } n = 0 \\ 1 & \text{falls } n = 1 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{sonst} \end{cases}$$

- [BONUS] Verwenden Sie den Stepper von DrScheme, um zu beobachten, wie `(fib 4)` berechnet wird. Wie Sie sehen, wird der Wert für `(fib 2)` bei der Berechnung von `(fib 4)` zweimal berechnet. Schreiben Sie nun eine Funktion `fib-fast`. Diese Funktion soll ebenfalls die Fibonacci-Zahl berechnen, allerdings diese Mehrfachberechnungen nicht durchführen. Außerdem soll die Funktion `fib-fast` endrekursiv sein.

Abgabe: elektronisch als `fib.scm`.