

## Hinweise zur Abgabe

Bitte reichen Sie Ihre Abgaben bis zum 8.1.2009 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, 6 Punkte]*

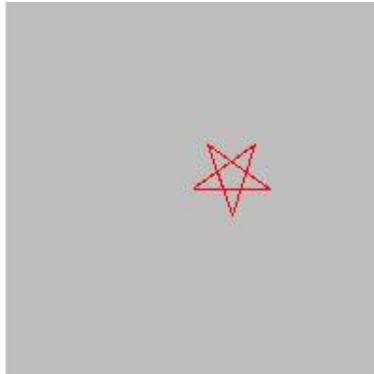
- Schreiben Sie, analog zu `list-fold`, ein Prozedur `tree-fold-inorder` mit dem Vertrag `(: btree-fold-inorder (number (number number -> number) btree -> number))`. Das erste Argument ist das neutrale Element für die übergebene Prozedur (zweites Argument). Die übergebene Prozedur muss durch die Eigenschaft des Baums (zwei Kinder) doppelt angewendet werden!
- Erzeugen Sie mit Hilfe `tree-fold-inorder` eine Prozedur `tree-sum-elem`, die für einen Baum mit numerischen Einträgen die Summe der Knoten berechnet, und `tree-mult-elem`, die die Elemente eines Baumes multipliziert.
- Schreiben Sie eine Prozedur `list-map` mit dem Vertrag `(: list-map ((list %a) (%a -> %b) -> (list %b)))`. `list-map` wendet dabei die gegebene Prozedur (zweites Argument) auf alle Elemente der Liste an und gibt eine neue Liste mit entsprechend modifizierten Einträgen zurück.
- Schreiben Sie analog eine Prozedur `tree-map` mit dem Vertrag `(: tree-map (btree (number -> number) -> btree))`.

*Abgabe:* Datei `higherorder.scm`.

## 2 Aufgabe *[Sprache: Die Macht der Abstraktion, Teachpack: turtle.ss, 5 Punkte]*

Lesen Sie das Tutorial für das auf der Informatik I Homepage verfügbar Teachpack `turtle.ss`. In dem Tutorial wird Ihnen erklärt, wie Sie mithilfe der Turtle Grafik ein Quadrat zeichnen.

- Schreiben Sie eine Funktion `star` welche einen fünfzackigen Stern malt. Der Stern sieht folgendermaßen aus:



Als Parameter soll die Funktion die Länge einer Linie des Sterns erhalten.

**Tipp:** Die Innenwinkel der Sternspitzen betragen 36 Grad.

- Zeichnen Sie den Stern mehrfach nacheinander, bis sie eine Schneeflocke erhalten.

*Abgabe:* Datei `star.scm`.

## 3 Aufgabe *[Sprache: Die Macht der Abstraktion, Teachpack: turtle.ss, 9 Punkte]*

- (a) Erstellen Sie strikt nach Konstruktionsanleitung eine Record-Definition für einen Punkt mit dem Namen `point`. Ein Punkt wird dabei durch seine x- und y-Koordinaten repräsentiert und hat zusätzlich eine Farbe.

- (b) Schreiben Sie eine Prozedur `plot-points` mit folgendem Vertrag:

```
(: plot-points ((list point) number number -> image)).
```

Die Prozedur `plot-points` soll die Liste der Punkte abarbeiten und mit Hilfe der Turtle jeweils zum nächsten Punkt eine Linie mit der definierten Farbe zeichnen. Die zwei Zahlen in der Parameterliste geben die Größe des Bildes an. Der Startpunkt der Turtle soll dabei immer (0,0) sein. Wie im Tutorial beschrieben ist die Blickrichtung anfangs nach rechts. Auf der Homepage finden Sie zwei Beispiellisten um Ihr Programm zu testen. Schreiben Sie dennoch ausreichend eigene Testfälle wie in der Vorlesung gelernt! Lagern Sie Funktionen in eigene Prozeduren aus, wo Sie es für sinnvoll halten und testen Sie diese ebenfalls.

**Tipp:** Die vorhandene Funktion `atan` kann hilfreich sein.

Bonus 1 (**5 Bonuspunkte**) Erstellen Sie aus den gegebenen Punktlisten ein Bild mit dem Haus und nebenan dem Tannenbaum. Dabei sollen natürlich nicht von Hand neue Punktlisten generiert werden!

**Tipp:** Der Hintergrund der Turtle-Grafik hat die Farbe `(make-color 190 190 190)`.

Bonus 2 (**3 Bonuspunkte**) Erweitern Sie Ihr Bild um mindestens eine Schneeflocke.

Bonus 3 (**keine Punkte**) Drucken Sie ihr Bild und schreiben Sie Ihrem Dozenten eine schöne Weihnachts bzw. Neujahrs Postkarte.

Georges-Köhler-Allee, Gebäude 079  
D-79110 Freiburg i. Br.

*Abgabe:* Datei `plot.scm`.