

Hinweise zur Abgabe

Bitte reichen Sie Ihre Abgaben bis zum 05.11.2009 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, d.h.:

1. Geben Sie den Vertrag an. **Falls der Vertrag fehlt, gibt die Aufgabe 0 Punkte.**
2. Wählen Sie abhängig vom Vertrag das richtige Funktionsgerüst aus.
3. Geben Sie Testfälle an.
4. Schreiben Sie den Funktionsrumpf. Dieser Schritt gliedert sich in weitere Unterschritte, die Sie entsprechend auswählen müssen. **Sie erhalten für diesen Teil der Aufgabe nur Punkte, wenn Sie 1.-3. befolgt haben.**

1 Aufgabe

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

Sei für $n \in \mathbb{N}$ die Funktion $f(n) = 1 + \dots + n = \sum_{i=1}^n i$ die Summe der Zahlen von 1 bis n .

- a. Berechnen Sie $f(1), f(2), f(3), f(5), f(10), f(11), f(15), f(150), f(15256)$.
Abgabe: Papier oder elektronisch als `aufgabe1-a.scm`.
- b. Finden Sie eine Formel F , so dass F aus drei primitiven Operatoren besteht und für alle $n \in \mathbb{N}$ gilt, dass $F(n) = f(n)$. Als primitive Operatoren zählen hier: $\{+, -, *, /\}$. Klammern können ebenfalls verwendet werden, um die Auswertungsreihenfolge festzulegen.
Abgabe: Papier
- c. Implementieren Sie die Funktion F als Scheme-Funktion. Folgen Sie den Konstruktionsanleitungen der Vorlesung.
Abgabe: elektronisch in der Datei `aufgabe1-c.scm`.
- d. [Bonus, 3 Punkte] Beweisen Sie, dass $F(n) = f(n)$ für alle $n \in \mathbb{N}$.
Abgabe: Papier

Hinweis: Es bietet sich an, die großen Werte von $f(n)$ mithilfe Ihres Scheme-Programms zu berechnen.

2 Aufgabe

[Sprache: Die Macht der Abstraktion - Anfänger, 5 Punkte]

Programmieren Sie eine Funktion `cone-volume`, die das Volumen eines Kegels berechnet. Die Funktion erhält die Höhe und den Radius des Grundkreises als Argumente. Verwenden Sie die Konstruktionsanleitung aus der Vorlesung!

- Erstellen Sie das Grundgerüst der Funktion `cone-volume` mit Vertrag, Beschreibung und mindestens drei Testfällen.

Abgabe: elektronisch in der Datei `aufgabe2-a.scm`.

- Füllen Sie den Funktionsrumpf mit einem Ausdruck, der das Volumen berechnet. Stellen Sie sicher, dass Ihre Testfälle erfolgreich sind.

Abgabe: elektronisch in der Datei `aufgabe2-b.scm`.

3 Aufgabe

[Sprache: Die Macht der Abstraktion - Anfänger, 8 Punkte]

Beobachten Sie die Auswertung der folgenden Ausdrücke im Stepper von DrScheme und geben Sie die Reduktionsregeln, die der Stepper ausführt, an!

Benennen Sie für die folgenden Ausdrücke für jeden Schritt des Steppers die Reduktionsregeln, die der Stepper angewendet hat.

Bsp.: $(+ (* 1 2) 3) \rightarrow (+ 2 3)$ „Applikation der eingebauten Prozedur (hier `*`)“

Hinweis: Es ist sinnvoll, die Regeln des Substitutionsmodells durczunummerieren, und dann die Regeln, die in einem Schritt des Steppers zur Anwendung kommen, durch ihre Nummer zu referenzieren.

```
(+ 1 (+ 1 (+ 1 0)))
```

```
(- (+ 2 3) (* 4 (cos 0)))
```

```
((lambda (a) a) (+ ((lambda (a) (+ a 2)) 3) 2))
```

```
(define pi 3.14159265)
```

```
(* 2 pi)
```

```
(define pi-quadrat (* pi pi))
```

```
(/ pi-quadrat 2)
```

```
(define quadrat
```

```
  (lambda (n)
```

```
    (* n n)))
```

```
(quadrat (+ 4 2))
```

```
((lambda (x) x) (lambda (x) x))
```

Abgabe: Papier.