
Informatik II<http://cis.informatik.uni-tuebingen.de/info2-ss-09>

Übungsblatt 1 (28.04.2009)

Abgabe: 05.05.2009, 15:00 Uhr

Hinweis zur Abgabe: Bitte reichen Sie Ihre Abgaben in elektronischer Form bis Dienstag, 05.05.2009, 15:00 Uhr, über die Homepage ein (unter „Übungen“ → „Aufgaben abgeben“) und geben Sie in der Pause der Dienstags-Vorlesung Ihre Papierabgaben ab.

Die Dateien für die digitale Abgabe müssen die in den Aufgaben angegebenen Dateinamen tragen. Stellen Sie sicher, dass in jeder Datei die Namen beider Teammitglieder stehen, bei Programmieraufgaben in Kommentaren. Nur ein Teammitglied muss die Dateien elektronisch abgeben. Programmieraufgaben müssen grundsätzlich immer elektronisch abgegeben werden, bei anderen Aufgaben und den Klassendiagrammen ist es optional. Wenn Sie die anderen Aufgaben elektronisch abgeben wollen, dann bitte ausschließlich im Textformat (*.txt), Klassendiagramme als .jpg oder .pdf. Geben Sie keine Word-Dokumente oder vergleichbare Formate ab!

Bitte kennzeichnen Sie Ihre Papierabgaben mit den Namen beider Teammitglieder und dem Namen Ihres Tutors. Heften Sie mehrseitige Abgaben zusammen und legen Sie Ihre Abgabe in die Mappe für Ihre Übungsgruppe. Die Mappen liegen in der Pause der Vorlesung aus.

**Verwenden Sie nur Sprachmittel, die in der Vorlesung behandelt wurden!
Benutzen Sie die Konstruktionsanleitungen!**

1. [6 Punkte] Betrachten Sie folgendes Java-Programm:

```
// Lied
public class Song {
    String artist;
    String title;
    int duration; // in Sekunden

    Song(String artist, String title, int duration) {
        this.artist = artist;
        this.title = title;
        this.duration = duration;
    }
}
```

- (a) [1 Punkt] Erstellen Sie mit BlueJ ein neues Projekt mit dem Namen **song**. Erstellen Sie in diesem Projekt eine neue Klasse, die den obigen Programmcode enthält.
- (b) [1 Punkt] Zeichnen Sie das zugehörige Klassendiagramm!
- (c) [1 Punkt] Erstellen Sie eine Beispielklasse mit dem Namen **Examples** für Beispielobjekte. Finden Sie drei Beispiele für die von den Objekten der Klasse repräsentierten Informationen und übersetzen Sie diese in Instanzen der Klasse!
- (d) [1 Punkt] Übersetzen Sie das Java-Programm in ein äquivalentes Scheme-Programm! Benutzen Sie die Konstruktionsanleitungen aus dem letzten Semester. Übertragen Sie auch Ihre Beispiele in das Scheme-Programm!
- (e) [1 Punkt] Was passiert, wenn Sie in Ihrer Beispielklasse folgende Zeile Java-Code hinzufügen und warum?

```
Song paradiseCity = new Song("Guns N' Roses", "Paradise City", 405.6);
```

- (f) [1 Punkt] Was passiert, wenn Sie den äquivalenten Scheme-Ausdruck in der REPL von DrScheme auswerten und warum?

Abgabe: Programme **song.jar** und **song.scm** und auf Papier

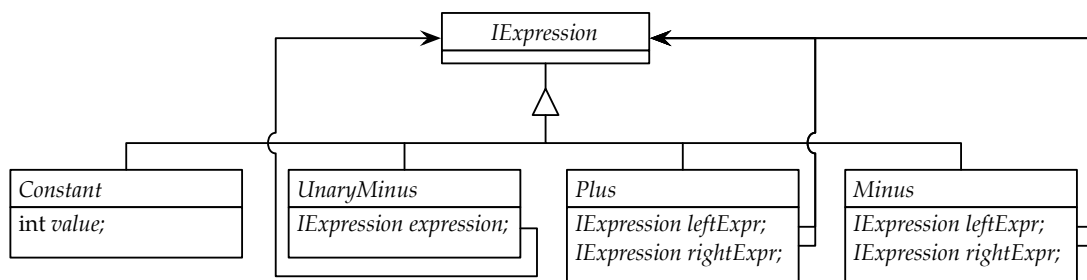
2. [6 Punkte] Am WSI gibt es (sehr vereinfacht) folgende Arten von Studierenden:

- Informatikstudierende haben einen Namen, ein Fachsemester, ein Nebenfach und ein Feld das aussagt, ob sie das Basispraktikum bereits absolviert haben.
- Bioinformatikstudierende haben einen Namen, ein Fachsemester und ein Feld das aussagt, ob sie das Tierphysiologiepraktikum bereits absolviert haben.
- Nebenfachstudierende haben einen Namen, ein Fachsemester und ein Hauptfach.

- (a) [2 Punkte] Implementieren Sie WSI-Studierende in DrScheme: Führen Sie eine Datenanalyse durch und schreiben Sie entsprechende Daten- und Recorddefinitionen.
- (b) [1 Punkt] Zeichnen Sie ein entsprechendes Klassendiagramm
- (c) [2 Punkte] Übersetzen Sie das Klassendiagramm in ein Java-Programm
- (d) [1 Punkt] Geben Sie in einer Beispielklasse fünf Beispiele für WSI-Studierende an.

Abgabe: Programme `wsistud.scm` und `wsistud.jar` und auf Papier

3. [6 Punkte] Schreiben Sie ein Programm für Ausdrücke in Java. Betrachten Sie folgendes Klassendiagramm:



Die Klasse `Constant` ist ein Ausdruck für ganze Zahlen, die Klasse `UnaryMinus` ist ein Ausdruck für den Vorzeichenwechsel eines Ausdrucks und die Klassen `Plus` und `Minus` sind Ausdrücke für zweistellige Operationen.

- (a) [2 Punkte] Übersetzen Sie das Klassendiagramm in die zugehörigen Klassendefinitionen.
- (b) [1 Punkt] Schreiben Sie eine Beispielklasse und stellen Sie folgenden Ausdruck als Beispiel dar:
 $4 + (-3) - 1 - (-(42))$
Dabei ist (-3) eine Konstante und $(-(42))$ eine positive Konstante mit Vorzeichenwechsel.
Schreiben Sie zwei weitere Beispiele!
- (c) [2 Punkte] Erfinden Sie eine beliebige dreistellige Operation. Erweitern Sie das Klassendiagramm entsprechend und übersetzen Sie die Erweiterung in Ihr Java-Programm.
- (d) [1 Punkt] Die Zahl 42 kann als Ausdruck unterschiedlich dargestellt werden, zum Beispiel als $12 + 30$. Finden Sie drei verschiedene Darstellungen und benutzen Sie dabei jede Operation (auch Ihre dreistellige) mindestens einmal.

Abgabe: Programm `expression.jar` und auf Papier

4. [8 Punkte] Bei der Bahn gibt es verschiedene Arten von Eisenbahnwagen zur Personenbeförderung:

- Personenwagen sind entweder erster oder zweiter Klasse, haben eine Anzahl von Sitzen, sind entweder einfach oder doppelstöckig und sind entweder als Großraum-, Abteil- oder als Mischwagen (Hälfte Großraum, Hälfte Abteil) aufgebaut. Benutzen Sie für den Aufbau der Wagen eine Aufzählung mit `enum`.
- Speisewagen haben eine Anzahl Tische und eine Anzahl Sitze. Die Anzahl der Sitze sind immer das Vierfache der Anzahl der Tische. Sorgen Sie dafür, dass es nicht möglich ist, Speisewagen mit einem ungültigen Verhältnis von Tischen und Sitzen zu erzeugen.
- Schlafwagen haben eine Anzahl Betten.

Schreiben Sie ein Java-Programm für Eisenbahnwagen:

- (a) [1 Punkt] Entwerfen und zeichnen Sie ein Klassendiagramm für Eisenbahnwagen.
- (b) [3 Punkte] Übersetzen Sie das Klassendiagramm in eine Klassendefinition.
- (c) [1 Punkt] Erstellen Sie eine Beispielklasse und entwickeln Sie für jede Klasse, die Sie geschrieben haben, drei Beispiele für die von den Objekten der Klasse repräsentierten Informationen und übersetzen Sie diese in Instanzen der Klasse!
- (d) [2 Punkte] Schreiben Sie Klassen für Züge, die aus einer beliebigen Anzahl von Wagen bestehen können. Erweitern Sie das Klassendiagramm entsprechend!
- (e) [1 Punkt] Erweitern Sie die Beispielklasse um drei Beispiele für Züge.

Abgabe: Programm `railcars.jar` und auf Papier

5. [4 Punkte] Reduzieren Sie folgende Terme in $\mathcal{L}_{\lambda A}$ zu Werten. Kennzeichnen Sie in jedem Reduktionsschritt, ob Sie eine β - oder eine δ -Reduktion vornehmen. Unterstreichen Sie dabei in jedem Reduktionsschritt alle vorhandenen β -Redexe:

- (a) $(+ ((\lambda x. (\lambda y. y) x) (- 2 1)) 8)$
- (b) $(\lambda x. (! x)) ((\lambda xy. (\&\& x y)) ((\lambda xy. (> x y)) 23 42) true)$

Dabei steht `!` für das boolesche `not` und `&&` für das boolesche `and`.

Abgabe: Auf Papier