
Softwaretechnik

<http://swt.informatik.uni-freiburg.de/node/94>
<http://proglang.informatik.uni-freiburg.de/teaching/swt/2008/>

Übungsblatt 1

2008-05-02

Aufgabe 1 (Javascript; (1+1+1) Punkte)

Sei folgender Javascript Code gegeben:

```
s = "some string";  
s.x = 13;  
s.x;
```

- (a) Laden Sie den Javascript Interpreter Rhino von der URL

<http://www.mozilla.org/rhino/download.html>

herunter und benutzen Sie Rhino um den Code auszuführen. (Der Interpreter wird mittels des Befehls `java -jar js.jar` gestartet, wobei die Datei `js.jar` in der heruntergeladenen `.zip` Datei enthalten ist.)

Welches Ergebnis gibt Rhino aus?

- (b) Ändern Sie die erste oder zweite Zeile so, dass für die dritte Zeile (`s.x`) nun 13 ausgegeben wird.
- (c) Erklären Sie dieses merkwürdige Verhalten. Wie würden Sie solche Fehler verhindern?

Aufgabe 2 (Typen für JAUS; (1+1+1+1+1) Punkte)

Welche der folgenden JAUS Ausdrücke sind typkorrekt? Geben Sie einen entsprechenden Herleitungsbaum für jeden typkorrekten Ausdruck an.

- (a) `1 + false`
- (b) `13 + (47 + 11)`
- (c) `!(true)`
- (d) `z + x`
- (e) `!z`

(In (d) und (e) soll `x` den Typ `int` und `z` den Typ `boolean` haben.)

Aufgabe 3 (Evaluierung von JAUS; (2+1) Punkte)

Werten Sie die folgenden JAUS Ausdrücke so weit wie möglich aus.

(a) $13 + (47 + 11)$

(b) $(1 + 1) + \text{false}$

Welche der resultierenden Ausdrücke sind Werte?

Aufgabe 4 (Typkorrektheit; 8 Punkte)

Beweisen Sie folgendes Theorem:

Falls $\vdash e_0 : t$ dann gibt es einen Werte e_n mit $\vdash e_n : t$ und

$$e_0 \longrightarrow e_1 \longrightarrow e_2 \longrightarrow \dots \longrightarrow e_{n-1} \longrightarrow e_n \quad .$$

Hinweis: Nachfolgendes Lemma ist hilfreich. Sie brauchen es nicht zu beweisen.

Lemma 1 (Normalisierung). *Zu jedem Ausdruck e_0 gibt es einen Ausdruck e_n so dass*

$$e_0 \longrightarrow e_1 \longrightarrow e_2 \longrightarrow \dots \longrightarrow e_{n-1} \longrightarrow e_n$$

und es gibt keinen Ausdruck e_{n+1} mit $e_n \longrightarrow e_{n+1}$.

Aufgabe 5 (Featherweight Java; 3 Punkte)

Sei folgendes Featherweight Java Programm gegeben:

```
class Author extends Object {
    String firstName;
    String lastName;

    Author(String firstName, String lastName) {
        super();
        this.firstName = firstName;
        this.lastName = lastName;
    }
}
```

```
class Book extends Object {
    Author author;

    Book(Author author) {
        this.author = author;
    }

    String getAuthorLastName() {
        return this.author.lastName;
    }
}
```

(Wir erweitern Featherweight Java großzügig um Unterstützung für Strings: Die Klasse `String` ist der Typ von Stringliteralen der Form "Dies ist ein String".)

Werten Sie jetzt den Ausdruck

```
new Book(new Author("Benjamin", "Pierce")).getAuthorLastName()
```

aus. Geben Sie alle Zwischenresultate an und erklären Sie für jeden Reduktionsschritt, welche Reduktionsregel Sie verwendet haben.

Abgabe: 2008-05-09, 12 Uhr vor der Saalübung im HS 00-036, Geb. 101.