3. Übungsblatt zu Programmiersprachenkonzepte

Manuel Schwarz, Michael Stypa

11. November 2010

Aufgabe 3.1

a) Korrigiertes Java-Programm

```
1  import java.io.*;
2  public class aufg1cor {
3    public static void main(String[] args){
4        if(args.length != 1) {
5        System.err.println("Aufruf: java aufg3_1 <Datei oder Verzeichnis>");
6        System.exit(0);
7      }
8      try {
9        delete(args[0]);
10     }
11     catch (IllegalArgumentException e) {
12       System.err.println(e.getMessage());
13     }
14   }
15
16   public static void delete(String datname) {
17     File f = new File(datname);
18     if (!f.exists()) fail("aufg3_1: Datei existiert nicht: " + datname);
19     if (!f.canWrite()) fail("aufg3_1: schreibgeschuetzt: " + datname);
20     if (f.isDirectory()) {
21         String[] dateien = f.list();
22           if (dateien.length > 0)
23         fail("aufg3_1: Verzeichnis nicht leer:" + datname);
24     }
25     boolean erfolg = f.delete();
26     if(!erfolg) fail("aufg3_1: Loeschen fehlgeschlagen");
27     }
28
29   protected static void fail(String s) throws IllegalArgumentException {
30     throw new IllegalArgumentException(s);
31   }
32 }
```

- Z. 3 Sting() args → String[] args; wird erkannt (Syntax)
- Z. 5 Anführungszeichen vor der letzten Klammer, wird erkannt (Syntax)
- Z. 16 Die delete()-Methode muss static sein, wird erkannt (Semantisch)
- Z. 18 f.exists() → !f.exists(), wird nicht erkannt
- Z. 20 schließende Klammer der if-Anweisung, wird erkannt (Syntax)
- Z. 21 ":=" → "=", wird erkannt (Lexikalisch)

- Z. 21 Semikolon am Ende der Zeile, wird erkannt (Syntax)
- Z. 22 “<” → “>”, wird nicht erkannt
- Z. 25 int → boolean, wird erkannt (Semantisch)
- Z. 29 Die fail()-Methode muss static sein, wird erkannt (Semantisch)
- Z. 32 schließende Klammer der class, wird erkannt (Syntax)

b) Funktionalität

Die dem Programm übergebene Datei bzw. das dem Programm übergebene (leere) Verzeichnis wird gelöscht. Zunächst wird geprüft, ob genau eine Datei übergeben wurde. Wenn nicht bricht das Programm ab, ansonsten wird versucht diese zu löschen. Dazu wird die delete()-Methode mit dem übergebenen Parameter (Datei oder Verzeichnis) aufgerufen. Anschließend wird geprüft, ob die Datei existiert und ob sie evtl. schreibgeschützt ist. Wenn sie nun nicht existiert oder kein Schreibrecht besteht, wird eine Exception geworfen. Falls die eingelesene Datei ein Verzeichnis ist, wird überprüft, ob dieses leer ist und ansonsten eine Exception geworfen. Schließlich wird die Datei gelöscht, wobei der Benutzer bei einem Misserfolg mit Hilfe einer Exception benachrichtigt wird.

Aufgabe 3.2

Vorteile	**Nachteile**
Portabilität	Performance
Single Stepping debugging	

Aufgabe 3.3

ZAHL	Symbol
3-2-1-meins	Fehler / Symbol warum?
15,03	Fehler
-25	Integer
FLOATP	Symbol
STRING	Symbol
DREI10	Symbol
4+1	Fehler / Symbol warum?
66/4	Rational
-33/11	Rational

Aufgabe 3.4

(+ 1 2 3 4 5 6 7 8 9)	45
(+ -1 (- 3 1))	1
(- (+ 3 5) (* 2 4) (/ 12 9))	-4/3
(- (+ 3.0 5) (* 2 4) (/ 7 2))	-3.5

Aufgabe 3.5

(ATOM ())	T
(ODDP 5)	T
(SYMBOLP 6)	NIL
(EQUAL 3 3.0)	NIL
(NOT (NOT T))	T
(NUMBERP (SYMBOLP X))	NIL