Universität Mannheim
Lehrstuhl für Praktische Informatik IV
Professor Dr. W. Effelsberg

Praktische Informatik I
WS 1999/2000

Übungsblatt 4 **Ausgabe: Mi, 17.11.99** **Abgabe: Di, 23.11.99, 18 Uhr**

Aufgabe 1: Syntax und Semantik [3 Punkte]

(a) [2 Punkte]

Gegeben sind folgende Variablen:

int a= 1; byte b= 1; short c=1;

Welche der nachfolgenden Ausdrücke enthalten Fehler? Geben Sie bei jedem Fehler an, ob es sich um einen semantischen oder syntaktischen Fehler handelt. Geben Sie eine Vermutung an, wie der Ausdruck korrekt ausgesehen sollte!

(i) $5 * 7 \| 30\%12 - 39/3 - 3 * 12 - 2 = 0$
(ii) $a++ == a$
(iii) $c = 100000 + 3 * a * b * 1/a/b$
(iv) $1 + 5 == 6 == 5 + 1$

(b) [1 Punkt] Korrigieren Sie die folgenden deutschen Sätze und geben Sie an, wo die Semantik und wo die Syntax falsch ist!

(i) „Ein Elefant is ein lilafarbenes, großes Tier."
(ii) „Wir programmiere in Java, weil Java als total veraltete Programmiersprache total veraltet ist."

Aufgabe 2: Java-Syntax [8 Punkte]

Die Java-Syntax unterstützt die drei Wiederholungsanweisungen *for*, *while* und *do – while*. Die drei Schleifen sind durch folgende Syntax-Regeln definiert (siehe auch Schader/ Schmidt-Thieme, S.513ff):

While-Anweisung:
while (*Ausdruck*) *Anweisung*

Do-Anweisung:
do Anweisung while (*Ausdruck*);

For-Anweisung:
$for\ (For-Init_{opt}\ ;\ Ausdruck_{opt}\ ;\ For-Update_{opt})\ Anweisung$

For-Init:
Anweisungsausdrucks – Liste
Lokale Variablendeklaration

For-Update:
Anweisungsausdrucks – Liste

Anweisungsausdrucks-Liste:
Anweisungsausdruck
Anweisungsausdrucks – Liste , *Anweisungsausdruck*

Zeigen Sie die folgende Behauptung: Die in der Java-Syntax enthaltenen drei Wiederholungsanweisungen *for*, *while* und *do* sind äquivalent. Mit anderen Worten, eine While-Anweisung läßt sich auch mit Hilfe einer Do-Anweisung und einer For-Anweisung ausdrücken und umgekehrt.

Tip: Zeigen Sie zunächst die Äquivalenz von while und do, indem Sie die Funktion einer Do-Anweisung mit Hilfe einer While-Anweisung simulieren und umgekehrt. Zeigen Sie anschließend die Äquivalenz von while und for in analoger Art und Weise.

Aufgabe 3: Java-Programmierung (Abgabe per abox) [9 Punkte]

Ein bekanntes Rätsel ist das Damenproblem im Schach (C.F. Gauß, 1850): 8 Damen sind so auf das Spielbrett zu stellen, daß sie sich gegenseitig nicht schlagen können. Wir betrachten eine Vereinfachung des Problems mit 8 Türmen. Formal ausgedrückt haben wir eine 8×8 Matrix mit Elementen aus $\{0,1\}$. Eine 1 gibt an, daß auf einem Feld ein Turm steht, eine 0 gibt an, daß das Feld frei ist (s. Tabelle 1). Eine Matrix stellt eine Lösung des Turmproblems dar, wenn in jeder Zeile und in jeder Spalte höchstens eine 1 zu finden ist.

Entwerfen Sie ein Java-Programm (Application), das feststellt ob eine gegebene Matrix eine Lösung des Turmproblems darstellt.

1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0

Tabelle 1: Matrix für eine Turmstellung auf dem Schachbrett

Anmerkung: Benutzen Sie den Klassennamen `NTuermeTester`! Das Programm muß als Kommandozeilenparameter den Namen einer Datei erwarten, die eine 8×8 Matrix erwarten der obigen Form erwartet. Das Programm muß als Ergebnis „ja“ ausgeben, falls die Matrix eine korrekte Lösung des Turmproblems darstellt, andernfalls soll „nein“ausgegeben werden. Hinweise zur Ein-/Ausgabe in Java finden Sie auf den Webseiten[1] der Übung. Die Testdateien `ntuermetest1.txt`[2]. und `ntuermetest2.txt`[3]. finden Sie ebenfalls im Web. Hier ein Beispiel für einen Kommandozeilenaufruf und der Ergebnisausgabe des von Ihnen zu entwickelnden Programmes:

```
 > java NTuermeTester tuermetest1.txt
ja
```

[1] http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/ws199900/pi1/ueb/blatt4/io-beispiel.html

[2] http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/ws199900/pi1/ueb/blatt4/ntuermetest1.txt

[3] http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/ws199900/pi1/ueb/blatt4/ntuermetest2.txt