

Musterlösung zu Übungsblatt 2

Praktische Informatik I

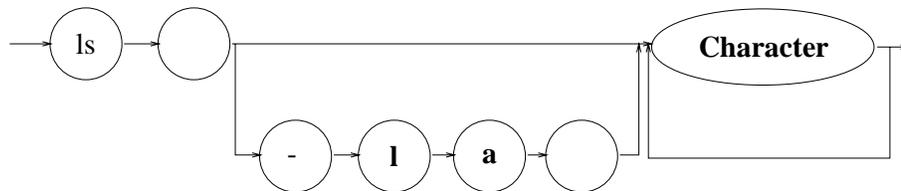
1. Dezember 1999

Im folgenden sind die Lösungen in kursiver Schrift angegeben, der eigentliche Aufgabentext in normaler Schrift.

1 Syntaxdiagramme

a)

Gegeben sei das folgende Syntaxdiagramm mit Character als die Buchstaben a-z und A-Z.



Geben Sie bei den folgenden Beispielen an, ob sie syntaktisch korrekt sind und wenn nicht, was falsch ist.

1. ls - source *falsch*: nach ls mit - müssen l und a als Zeichen folgen.
2. ls -la source *richtig*
3. ls -l-a source *falsch*: zwischen l und a darf kein - sein.
4. la source *falsch*: ls statt la.
5. ls source2 *falsch*: Character darf nur aus Zeichen a-z oder A-Z bestehen, nicht aus Zahlen.

b)

Schreiben Sie die folgenden Zahlen in korrekter Java-Syntax:

1. eine Zahl vom Typ long mit dem Wert 32 *32L*
2. eine Zahl vom Typ int mit dem Wert 32 *32*

Bewertungsvorschlag: Für jede richtig beantwortete Unteraufgabe einen Punkt.

c)

Welche der 6 Zahlen sind im falschen Format:

1. 034L *richtig, long, Oktalzahl*
2. 4587L *richtig, long*
3. 7.f *richtig, float*
4. .945D *richtig, double*
5. 33.33E12 *richtig, double*
6. 0.E+27 *richtig, double*

2 Java-Syntax

Das nachfolgende Java-Programm enthält eine Reihe syntaktischer Fehler. Nennen Sie diese Fehler und begründen Sie jeweils, warum es sich um einen Fehler handelt. Korrigieren Sie die Fehler. Welche Ausgabe erzeugt das korrigierte Programm?

Ein Beispiel soll die Aufgabe nochmals erläutern:

```
flott variable = 3e-1.2;
```

Diese Zeile enthält zwei Fehler. Erstens ist `flott` kein Java-Schlüsselwort, es müsste `float` heißen. Zweitens ist `3e-1.2` keine Float-Zahl, da der Exponent keinen Dezimalpunkt enthalten darf.

```
class Fehlertext {

/** Hier beginnt die Programmausführung: */
public static void main(String[] args) {

int start-wert = 1;
//int end-wert = 10; /*
Berechnung: */
int n=1; int m=10; int
$i = n+++++m; int 2 = $i;
if ((n&m)!=!$i) System.out.
println('Ergebnis '+$i+' '+n+' '+m)
;int k=7; k%=1==2?3:4;
System.out.println(k);
do { k>>>=.3e+1;
k<=0x2; } until ((n==m) !=!false);
if ( k>=3 ) k
<<=0x2; int
int2=$i;
int $1=2;}}
```

Die Fehler im Einzelnen:

- `int start-wert = 1;` *“Invalid declaration.” S.Seite 17 Schader: Ein Bezeichner muss mit einem Buchstaben beginnen und darf aus Unicode-Buchstaben (A-Z, a-z, _ und \$) oder Unicode-Ziffern (0-9) bestehen.*
- Berechnung: `*/` *“Missing term.” und andere Fehler. Durch den Zeilenkommentar in der vorhergehenden Zeile (“//”) wird die Kommentaranfangsmarkierung (“/*”) ignoriert, und die Kommentarendemarkierung erzeugt eine Fehlermeldung. Berechnung: wird als Label interpretiert.*
- `$i = n+++++m;` *“Invalid argument to ++.” Durch fehlende Leerzeichen wird die Kette von Pluszeichen falsch interpretiert. Wie Java die Zeile tatsächlich auswertet, ist mir nicht ganz klar, da “+” als Vorzeichen und “++” als Inkrement die gleiche Priorität haben. `n (+ (+ (+ ++ m)))` ist eine Möglichkeit, wobei die “+“-Blöcke von rechts nach links Inkrement, Vorzeichen, Operator und ein Problem sind...*
- `int 2 = $i;` *“Invalid expression statement.” Siehe erster Fehler, Bezeichner muss mit Buchstaben beginnen.*
- `if ((n&m)!=!$i) System.out.` *“Invalid left hand side of assignment.” “Incompatible type for !. Can’t convert int to boolean.” Das “=!” ist verdreht. Dadurch wird der Ausdruck als Zuweisung interpretiert, wobei das Ergebnis von “n&m” ein boolescher Ausdruck ist (true oder false) und keine Variable. Desweiteren kann man “!” nicht ohne weiteres auf Integerwerte anwenden.*
- `println(“Ergebnis ’+$i+“ ’+n+“ ’+m)` *“String not terminated at end of line.” Der String ist nicht korrekt terminiert.*
- `k<=0x2; }` *“Invalid expression statement.” Der <= Operator ist ein Vergleichsoperator, hier wird eine Zuweisung erwartet.*
- `do { k>>>=.3e+1;` *“Incompatible type for <<<=. Explicit cast needed to convert double to int.” Hier ist ein Fehler in der Fehlermeldung... Der Operator “<<<=” erwartet einen Integerwert, keinen Double.*
- `until ((n==m) !=!false);` *“while’ expected.” until ist kein Java - Schlüsselwort, while beendet eine do-Schleife.*

Die Ausgabe des korrigierten Programms ist

`Ergebnis 12 2 11`

`3`

Die Ergebnisse kommen wie folgt zustande:

- `Ergebnis 12 2 11:` `$i` wird das Ergebnis der Addition von `n (1)` und `++m (11)` zugewiesen. `n` und `m` werden beide um eins erhöht.
- `3:` `k` beinhaltet den Wert 7. `k` wird der Rest der Division von 7 und 4 zugewiesen. `7 / 4 = 1 Rest 3`