

## Vorlesung Grundlagen der Programmiersprachen Sommer 2004 - Übungsblatt 2

Ausgabe: 16.06.2004 -- Abgabe: 23.06.2004, 13:00 Uhr, im Abgabekasten auf dem D3-Flur.

### Aufgabe 3 (Programme in vier verschiedenen Programmiersprachen)

Die nachfolgenden vier Verzeichnisse enthalten Programme in vier verschiedenen Programmiersprachen:

- /homes/info-f/gdp/aufgaben/blatt2/lang/Fortran77
- /homes/info-f/gdp/aufgaben/blatt2/lang/Prolog
- /homes/info-f/gdp/aufgaben/blatt2/lang/sh
- /homes/info-f/gdp/aufgaben/blatt2/lang/SML

- a) Führen Sie diese vier Programme aus. Legen Sie dazu eine Kopie der Programme aus dem Verzeichnis lang an.

Weitere Hinweise dazu finden Sie in Usage im Verzeichnis  
/homes/info-f/gdp/aufgaben/blatt2/lang.

- b) Die vier Programme geben alle eine Liste der ganzen Zahlen von 15 absteigend bis 1 aus. Modifizieren Sie die Programme so, daß die erzeugte Ausgabe aufsteigend von 1 bis 15 erfolgt.

### Aufgabe 4 (Aussagen aus dem Java Reference Manual)

Ordnen Sie die folgenden Aussagen aus dem Java Reference Manual den vier Ebenen der Spracheigenschaften zu.

1. *A block is a sequence of statements and local variable declaration statements within braces.*
2. *A return statement with an expression must be contained in a method declaration that is declared to return a value or a compile-time error occurs.*
3. *If execution of the try block completes normally, then no further action is taken and the try statement completes normally.*
4. *The boolean type has two values, represented by the literals true and false, formed from ASCII letters.*
5. *A widening conversion of a signed integer value to an integral type T simply sign-extends the two's-complement representation of the integer value to fill the wider format.*
6. *It is a compile-time error for a line terminator to appear after the opening " and before the closing matching ".*
7. *The type of the operand expression of the unary - operator must be a primitive numeric type.*
8. *An array type is written as the name of an element type followed by some number of empty pairs of square brackets [ ].*

### Aufgabe 5 (Kontextfreie Grammatik verstehen)

\*\*\* Diese Aufgabe wird bewertet. \*\*\*

Im Folgenden ist rechts eine kontextfreie Grammatik für eine imperative Sprache angegeben.

a) Geben Sie die Terminalsymbole und Nichtterminalsymbole der Grammatik an.

b) Wie sieht der kleinste Ableitungsbaum (in Anzahl Knoten) zu dieser Grammatik aus?

c) Herr D. E. Tektiv stellt beim Lesen der Grammatik fest, dass die Grammatik mehrdeutig ist. Markieren Sie alle Produktionen, die Mehrdeutigkeiten herbeiführen. Schlagen Sie geeignete Maßnahmen vor, mit denen man die Mehrdeutigkeiten entfernen kann, ohne die Sprache zu verändern.

d) Geben Sie die Ableitungsschritte an, mit denen der folgende Satz aus dem Startsymbol abgeleitet werden kann:

```
main {  
    input(value);  
    if (value)  
        output(log(value));  
}
```

```
p1: prog      ::= 'main' stmtblock  
p2: stmtblock ::= '{' stmts '}'  
p3: stmtblock ::= stmt  
p4: stmts     ::= stmts stmt  
p5: stmts     ::= stmt  
p6: stmt      ::= app_id '=' expr ';'   
p7: stmt      ::= 'if' '(' expr ')' stmtblock  
                'else' stmtblock  
p8: stmt      ::= 'while' '(' expr ')' stmtblock  
p9: stmt      ::= 'if' '(' expr ')' stmtblock  
p10: stmt     ::= 'input' '(' app_id ')' ';'   
p11: stmt     ::= 'output' '(' app_id ')' ';'   
p12: app_id   ::= identifier  
p13: expr     ::= call  
p14: expr     ::= integer  
p15: expr     ::= app_id  
p16: call     ::= app_id '(' params ')'  
p17: params   ::= params ',' params  
p18: params   ::= param  
p19: param    ::= expr
```