

$$\langle letter \rangle \rightarrow [a - z]$$

$$| [A - Z]$$

$$\langle digit \rangle \rightarrow [0 - 9]$$

$$\langle program \rangle \rightarrow \langle roots \rangle$$

$$\langle roots \rangle \rightarrow \langle root \rangle$$

$$| \langle root \rangle \langle roots \rangle$$

$$\langle root \rangle \rightarrow \langle function \rangle$$

$$| \langle dcl \rangle;$$

$$\langle dcl \rangle \rightarrow \langle type \rangle \langle id \rangle$$

$$| \langle type \rangle \langle assign \rangle$$

$$\langle id \rangle \rightarrow \langle letter \rangle$$

$$| \langle id \rangle \langle letter \rangle$$

$$| \langle id \rangle \langle digit \rangle$$

$$\langle assign \rangle \rightarrow \langle id \rangle <- \langle expr \rangle$$

$$\langle type \rangle \rightarrow \langle primitive-type \rangle$$

$$| \langle array-type \rangle$$

$$\langle primitive-type \rangle \rightarrow \text{bool}$$

$$| \text{double}$$

$$| \text{int}$$

$$| \text{char}$$

$$\langle array-type \rangle \rightarrow \langle type \rangle [ ]$$

$$| \text{string}$$

$$\langle function \rangle \rightarrow \text{function } \langle id \rangle \text{ returns } \langle type \rangle \text{ using } (\langle params \rangle) \text{ begin } \langle stmts \rangle \text{ return } \langle expr \rangle;$$

$$\text{end}$$

$$| \text{function } \langle id \rangle \text{ returns nothing using } (\langle params \rangle) \text{ begin } \langle stmts \rangle \text{ return nothing; end}$$

$$\langle params \rangle \rightarrow \langle subparams \rangle$$

$$| \varepsilon$$

$$\langle subparams \rangle \rightarrow \langle type \rangle \langle id \rangle, \langle subparams \rangle$$

$$| \langle type \rangle \langle id \rangle$$

$$\langle stmts \rangle \rightarrow \langle stmt \rangle$$

$$| \langle stmt \rangle \langle stmts \rangle$$

$$\langle stmt \rangle \rightarrow \langle assign \rangle;$$

$$| \langle if \rangle$$

$$| \langle while \rangle$$

$$| \langle from \rangle$$

$$| \varepsilon$$

$$| \langle dcl \rangle;$$

$$| \langle functioncall \rangle;$$

$$| \langle switch \rangle$$

$\langle \text{switch} \rangle \rightarrow \text{switch } (\langle \text{expr} \rangle) \text{ begin } \langle \text{cases} \rangle \text{ end}$

$\langle \text{cases} \rangle \rightarrow \text{case } \langle \text{expr} \rangle: \langle \text{stmts} \rangle \langle \text{endcase} \rangle$

$\langle \text{endcase} \rangle \rightarrow \langle \text{cases} \rangle$

| break;  
| break;  $\langle \text{cases} \rangle$   
| default:  $\langle \text{stmts} \rangle$  break;  
| break; default:  $\langle \text{stmts} \rangle$  break;

$\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle$

|  $\langle \text{expr} \rangle - \langle \text{term} \rangle$   
|  $\langle \text{term} \rangle$

$\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle * \langle \text{factor} \rangle$

|  $\langle \text{term} \rangle / \langle \text{factor} \rangle$   
|  $\langle \text{factor} \rangle$

$\langle \text{factor} \rangle \rightarrow ( \langle \text{expr} \rangle )$

|  $\langle \text{id} \rangle$   
|  $\langle \text{plusminus} \rangle \langle \text{digit} \rangle$   
|  $\langle \text{plusminus} \rangle \langle \text{nummeric} \rangle$   
|  $\text{"} \langle \text{string} \rangle \text{"}$

$\langle \text{plusminus} \rangle \rightarrow \varepsilon$

| -

$\langle \text{nummeric} \rangle \rightarrow \langle \text{digit} \rangle$

|  $\langle \text{digit} \rangle \langle \text{nummeric} \rangle$   
|  $\cdot \langle \text{digitonly} \rangle$

$\langle \text{digitonly} \rangle \rightarrow \langle \text{digit} \rangle$

|  $\langle \text{digit} \rangle \langle \text{digitonly} \rangle$

$\langle \text{string} \rangle \rightarrow \langle \text{letter} \rangle$

|  $\langle \text{digit} \rangle$   
|  $\langle \text{symbols} \rangle$   
|  $\langle \text{symbols} \rangle \langle \text{string} \rangle$   
|  $\langle \text{digit} \rangle \langle \text{string} \rangle$   
|  $\langle \text{letter} \rangle \langle \text{string} \rangle$   
|  $\varepsilon$

$\langle \text{symbols} \rangle \rightarrow !$

| %  
| ^  
| &  
| \*  
| (  
| )  
| \_  
| +  
| |  
| ~

```

| -
| =
| ,
| {
| }
| [
| ]
| :
| ;
| ,
| ?
| ,
| .
| /
| ' ,

```

$\langle from \rangle \rightarrow \text{from } \langle expr \rangle \text{ to } \langle logexpr \rangle \text{ step } \langle expr \rangle \text{ begin } \langle stmts \rangle \text{ end}$

$\langle while \rangle \rightarrow \text{while}(\langle logexpr \rangle) \text{ begin } \langle stmts \rangle \text{ end}$

$\langle if \rangle \rightarrow \text{if}(\langle logexpr \rangle) \text{ begin } \langle stmts \rangle \langle endif \rangle$

$\langle endif \rangle \rightarrow \text{end else } \langle if \rangle$   
 |   end else begin  $\langle stmts \rangle$  end  
 |   end

$\langle logexpr \rangle \rightarrow \langle logexpr \rangle \text{ OR } \langle andcomp \rangle$   
 |    $\langle andcomp \rangle$

$\langle andcomp \rangle \rightarrow \langle andcomp \rangle \text{ AND } \langle comp \rangle$   
 |    $\langle comp \rangle$

$\langle comp \rangle \rightarrow \langle boolean-operand \rangle \langle comparison-operator \rangle \langle boolean-operand \rangle$

$\langle boolean-operand \rangle \rightarrow \text{true}$   
 |   false  
 |    $\langle expr \rangle$   
 |    $\langle boolean \rangle$

$\langle boolean \rangle \rightarrow \text{!}(\langle logexpr \rangle)$   
 |    $(\langle logexpr \rangle)$

$\langle comparison-operator \rangle \rightarrow >$   
 |   <  
 |   <=  
 |   >=  
 |   !=  
 |   =

$\langle functioncall \rangle \rightarrow \text{call } \langle id \rangle(\langle params \rangle)$