⟨*letter*⟩ → [a - z]
| [A - Z]

⟨*digit*⟩ → [0 - 9]

⟨*program*⟩ → ⟨*roots*⟩

⟨*roots*⟩ → ⟨*root*⟩
| ⟨*root*⟩ ⟨*roots*⟩

⟨*root*⟩ → ⟨*function*⟩
| ⟨*dcl*⟩;

⟨*dcl*⟩ → ⟨*type*⟩ ⟨*id*⟩
| ⟨*type*⟩ ⟨*assign*⟩

⟨*id*⟩ → ⟨*letter*⟩
| ⟨*id*⟩ ⟨*letter*⟩
| ⟨*id*⟩ ⟨*digit*⟩

⟨*assign*⟩ → ⟨*id*⟩ `<-` ⟨*expr*⟩

⟨*type*⟩ → ⟨*primitive-type*⟩
| ⟨*array-type*⟩

⟨*primitive-type*⟩ → bool
| double
| int
| char

⟨*array-type*⟩ → ⟨*type*⟩ [ ]
| string

⟨*function*⟩ → function ⟨*id*⟩ returns ⟨*type*⟩ using (⟨*params*⟩) begin ⟨*stmts*⟩ return ⟨*expr*⟩; end
| function ⟨*id*⟩ returns nothing using (⟨*params*⟩) begin ⟨*stmts*⟩ return nothing; end

⟨*params*⟩ → ⟨*subparams*⟩
| ε

⟨*subparams*⟩ → ⟨*type*⟩ ⟨*id*⟩, ⟨*subparams*⟩
| ⟨*type*⟩ ⟨*id*⟩

⟨*stmts*⟩ → ⟨*stmt*⟩
| ⟨*stmt*⟩⟨*stmts*⟩

⟨*stmt*⟩ → ⟨*assign*⟩;
| ⟨*if*⟩
| ⟨*while*⟩
| ⟨*from*⟩
| ε
| ⟨*dcl*⟩;
| ⟨*functioncall*⟩;
| ⟨*switch*⟩

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

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

$\langle \mathit{endcase} \rangle \rightarrow \langle \mathit{cases} \rangle$
| break;
| break; $\langle \mathit{cases} \rangle$
| default: $\langle \mathit{stmts} \rangle$ break;
| break; default: $\langle \mathit{stmts} \rangle$ break;

$\langle \mathit{expr} \rangle \rightarrow \langle \mathit{expr} \rangle + \langle \mathit{term} \rangle$
| $\langle \mathit{expr} \rangle$-$\langle \mathit{term} \rangle$
| $\langle \mathit{term} \rangle$

$\langle \mathit{term} \rangle \rightarrow \langle \mathit{term} \rangle$ * $\langle \mathit{factor} \rangle$
| $\langle \mathit{term} \rangle$ / $\langle \mathit{factor} \rangle$
| $\langle \mathit{factor} \rangle$

$\langle \mathit{factor} \rangle \rightarrow$ ($\langle \mathit{expr} \rangle$)
| $\langle \mathit{id} \rangle$
| $\langle \mathit{plusminus} \rangle \langle \mathit{digit} \rangle$
| $\langle \mathit{plusminus} \rangle \langle \mathit{nummeric} \rangle$
| "$\langle \mathit{string} \rangle$"
| $\langle \mathit{functioncall} \rangle$

$\langle \mathit{plusminus} \rangle \rightarrow \varepsilon$
| -

$\langle \mathit{nummeric} \rangle \rightarrow \langle \mathit{digit} \rangle$
| $\langle \mathit{digit} \rangle \langle \mathit{nummeric} \rangle$
| .$\langle \mathit{digitonly} \rangle$

$\langle \mathit{digitnonly} \rangle \rightarrow \langle \mathit{digit} \rangle$
| $\langle \mathit{digit} \rangle$ $\langle \mathit{digitonly} \rangle$

$\langle \mathit{string} \rangle \rightarrow \langle \mathit{letter} \rangle$
| $\langle \mathit{digit} \rangle$
| $\langle \mathit{symbols} \rangle$
| $\langle \mathit{symbols} \rangle$ $\langle \mathit{string} \rangle$
| $\langle \mathit{digit} \rangle$ $\langle \mathit{string} \rangle$
| $\langle \mathit{letter} \rangle$ $\langle \mathit{string} \rangle$
| ε

$\langle \mathit{symbols} \rangle \rightarrow$!
| %
| ˆ
| &
| *
| (
|)
| _
| +
| |

```
  |  ˜
  |  -
  |  =
  |  `
  |  {
  |  }
  |  [
  |  ]
  |  :
  |  ;
  |  ’
  |  ?
  |  ,
  |  .
  |  /
  |  ’ ’
  |

⟨from⟩ → from ⟨expr⟩ to ⟨logexpr⟩ step ⟨expr⟩ begin ⟨stmts⟩ end

⟨while⟩ → while(⟨logexpr⟩) begin ⟨stmts⟩ end

⟨if⟩ → if(⟨logexpr⟩) begin ⟨stmts⟩ ⟨endif⟩

⟨endif⟩ → end else ⟨if⟩
  |  end else begin ⟨stmts⟩ end
  |  end

⟨logexpr⟩ → ⟨logexpr⟩ OR ⟨andcomp⟩
  |  ⟨andcomp⟩

⟨andcomp⟩ → ⟨andcomp⟩ AND ⟨comp⟩
  |  ⟨comp⟩

⟨comp⟩ → ⟨boolean-operand⟩ ⟨comparison-operator⟩ ⟨boolean-operand⟩

⟨boolean-operand⟩ → true
  |  false
  |  ⟨expr⟩
  |  ⟨boolean⟩

⟨boolean⟩ → !(⟨logexpr⟩)
  |  (⟨logexpr⟩)

⟨comparison-operator⟩ → >
  |  <
  |  <=
  |  >=
  |  !=
  |  =

⟨functioncall⟩ → call ⟨id⟩(⟨params⟩)
```