```
start : program;
            program :
                         program unit
                         \underline{\text{unit}}
               unit : var_declaration
                         \underline{\text{func\_declaration}}
                         func\_definition
    func_declaration :
                         type_specifier ID LPAREN parameter_list RPAREN SEMICOLON
                         type_specifier ID LPAREN RPAREN SEMICOLON
      func_definition
                         type_specifier ID LPAREN parameter_list RPAREN
                         compound_statement
                         type_specifier ID LPAREN RPAREN compound_statement
      parameter_list
                         parameter_list_COMMA_type_specifier_ID
                         parameter_list COMMA type_specifier
                         \underline{\text{type\_specifier ID}}
                         type_specifier
                         LCURL statements RCURL
compound_statement
                         LCURL RCURL
     var_declaration
                         type_specifier declaration_list SEMICOLON
       type_specifier
                         INT
                         FLOAT
                         VOID
     declaration_list :
                         declaration_list COMMA ID
                         declaration_list COMMA ID LTHIRD CONST_INT RTHIRD
                         ID LTHIRD CONST_INT RTHIRD
         statements : statement
                         statements statement
```

statement : var\_declaration

 $\frac{expression\_statement}{compound\_statement}$ 

FOR LPAREN expression\_statement expression\_statement expression

RPAREN statement

IF LPAREN expression RPAREN statement

IF LPAREN expression RPAREN statement ELSE statement

WHILE LPAREN expression RPAREN statement PRINTLN LPAREN ID RPAREN SEMICOLON

RETURN expression SEMICOLON

;

 $expression\_statement : SEMICOLON$ 

expression SEMICOLON

variable : ID

ID LTHIRD expression RTHIRD

;

expression:  $logic_expression$ 

variable ASSIGNOP logic\_expression

;

logic\_expression : rel\_expression

rel\_expression LOGICOP rel\_expression

;

rel\_expression : simple\_expression

simple\_expression RELOP simple\_expression

;

simple\_expression : term

simple\_expression ADDOP term

term : unary\_expression

term MULOP unary\_expression

;

unary\_expression : ADDOP unary\_expression

NOT unary\_expression

 $\underline{factor}$ 

;

<u>factor</u> : va<u>riable</u>

ID LPAREN argument\_list RPAREN

LPAREN expression RPAREN

CONST\_INT CONST\_FLOAT

 $v\underline{ariable\ IN}COP$ 

variable DECOP

:

 $argument\_list$  : arguments

.

arguments : arguments COMMA logic\_expression

logic\_expression

;