```
          start    :    program;

        program    :    program unit
                   |    unit
                   ;

           unit    :    var_declaration
                   |    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
                   |    type_specifier ID
                   |    type_specifier
                   ;

compound_statement :    LCURL statements RCURL
                   |    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
                   |    ID LTHIRD CONST_INT RTHIRD
                   ;

      statements   :    statement
                   |    statements statement
                   ;
```

$$
\begin{array}{rcl}
\text{statement} & : & \text{var\_declaration} \\
& | & \text{expression\_statement} \\
& | & \text{compound\_statement} \\
& | & \text{FOR LPAREN expression\_statement expression\_statement expression} \\
& & \text{RPAREN statement} \\
& | & \text{IF LPAREN expression RPAREN statement} \\
& | & \text{IF LPAREN expression RPAREN statement ELSE statement} \\
& | & \text{WHILE LPAREN expression RPAREN statement} \\
& | & \text{PRINTLN LPAREN ID RPAREN SEMICOLON} \\
& | & \text{RETURN expression SEMICOLON} \\
& ; & \\
\\
\text{expression\_statement} & : & \text{SEMICOLON} \\
& | & \text{expression SEMICOLON} \\
& ; & \\
\\
\text{variable} & : & \text{ID} \\
& | & \text{ID LTHIRD expression RTHIRD} \\
& ; & \\
\\
\text{expression} & : & \text{logic\_expression} \\
& | & \text{variable ASSIGNOP logic\_expression} \\
& ; & \\
\\
\text{logic\_expression} & : & \text{rel\_expression} \\
& | & \text{rel\_expression LOGICOP rel\_expression} \\
& ; & \\
\\
\text{rel\_expression} & : & \text{simple\_expression} \\
& | & \text{simple\_expression RELOP simple\_expression} \\
& ; & \\
\\
\text{simple\_expression} & : & \text{term} \\
& | & \text{simple\_expression ADDOP term} \\
& ; & \\
\text{term} & : & \text{unary\_expression} \\
& | & \text{term MULOP unary\_expression} \\
& ; & \\
\\
\text{unary\_expression} & : & \text{ADDOP unary\_expression} \\
& | & \text{NOT unary\_expression} \\
& | & \text{factor} \\
& ; &
\end{array}
$$

```
factor          :   variable
                |   ID LPAREN argument_list RPAREN
                |   LPAREN expression RPAREN
                |   CONST_INT
                |   CONST_FLOAT
                |   variable INCOP
                |   variable DECOP
                ;

argument_list   :   arguments
                |
                ;

arguments       :   arguments COMMA logic_expression
                |   logic_expression
                ;
```