```
#------------------------------------------------------------
# MOHAMMADREZA KHOSRAVIAN
# SAM KHANAKI
# ALI SADEGHI
#
# GRAMMAR RULES FOR LANGUAGE IMPLEMENTATION
# SHARIF UNIVERSITY OF TECHNOLOGY - SPRING 2025
#------------------------------------------------------------

Program → StatementList

StatementList → Statement StatementList
              | ε

Statement → VariableDeclaration
          | FunctionDeclaration
          | Assignment
          | IfStatement
          | LoopStatement
          | Block
          | FunctionCall
          | PrintStatement
          | ReturnStatement
          | BreakStatement
          | ContinueStatement

BreakStatement → 'break' ';'

ContinueStatement → 'continue' ';'

Block → '{' StatementList '}'

VariableDeclaration → 'var' Identifier ':' Type '=' Expression ';'

Type → 'int'
     | 'float'
     | 'string'
     | 'bool'
     | 'list' '[' Type ']'
     | 'nulltype'

FunctionDeclaration → 'func' Identifier '(' ParameterList ')' ':' Type Block

ParameterList → Parameter ',' ParameterList
              | Parameter
              | ε

Parameter → Identifier ':' Type

Assignment → Identifier '=' Expression ';'

IfStatement → MatchedIf
            | UnmatchedIf

MatchedIf → 'if' '(' Expression ')' Block 'else' Block

UnmatchedIf → 'if' '(' Expression ')' Block
            | 'if' '(' Expression ')' MatchedIf 'else' UnmatchedIf
```

```
LoopStatement → WhileLoop
              | ForLoop

WhileLoop → 'while' '(' Expression ')' Block

ForLoop → 'for' '(' ForAssignment ';' Expression ';' ForAssignment ')' Block

ForAssignment → Identifier '=' Expression

ArgumentList → Expression ',' ArgumentList
             | Expression
             | ε

FunctionCall → Identifier '(' ArgumentList ')'
             | Identifier '(' ArgumentList ')' ';'

PrintStatement → 'print' '(' Expression ')' ';'

ReturnStatement → 'return' Expression ';'

Expression → SimpleExpression
           | SimpleExpression ComparativeOperator Expression

SimpleExpression → Term
                 | SimpleExpression LogicalOperator Term

Term → Factor
     | Term ArithmeticOperator Factor

Factor → Identifier
       | Literal
       | '(' Expression ')'
       | FunctionCall
       | UnaryOperator Factor
       | 'NULL'

Literal → IntegerLiteral
        | FloatLiteral
        | StringLiteral
        | ListLiteral
        | 'true'
        | 'false'

ListLiteral → '[' ExpressionList ']'

ExpressionList → Expression ',' ExpressionList
              | Expression
              | ε

ComparativeOperator → '<'
                    | '>'
                    | '<='
                    | '>='
                    | '=='
                    | '!='

LogicalOperator → '&&'
                | '||'
```

```
ArithmeticOperator → '+'
                   | '-'
                   | '*'
                   | '/'
                   | '%'

UnaryOperator → '-'
              | '!'
```