<letter> ::= a | ... | z | A | B | ... | Z

<digit> ::= 0 | ... | 9

<underscore> ::= _

<operator> ::= + | - | * | / | = | == | < | <= | > | >= | !=

<separator> ::= () | {} | ; | space | ,

<reserved_word> ::= Integer | String | ArrayList | Do | While | For | Const | Program | Const | Read | Print

<identifier> ::= <letter> | <letter><identifier> | <underscore><identifier> |

<letter><sequenceOfCharacters> | <underscore><sequenceOfCharacters>

<sequenceOfCharacters> ::= <letter> | <digit>

<no> ::= <digit> | <digit><no>

<noconst> ::= +<no> | -<no> | <no>

<character> ::= '<letter>' | ' <digit>' | ' <underscore>'

<char> ::= <letter> | <digit>

<string> ::= <char><string>

<conststring> ::= "<string>"

<program> ::= "VAR" <decllist> ";" <cmpdstmt> "."

<decllist> ::= <declaration> | <declaration> ";" <decllist>

<declaration> ::= <identifier> ":" <type>

<type1> ::= boolean | char | Integer | Float

<listOfDeclarations> ::= "Array" "[" nr "]" "Of" <type1>

<type> ::= <type1> | <listOfDeclarations>

<cmpdstmt> ::= "BEGIN" <statementList> "END"

<statementList> ::= <statement> | <statement> ";" <statementList>

<statement> ::= <simpleStatement> | <structuredStatement>

<simpleStatement> ::= <assignStatement> | <IOStatement>

<assignStatement> ::= <identifier> "=" <expression>

<expression> ::= <expression> "+" <term> | <term> | <expression> "-" <term>

<term> ::= <term> "*" <factor> | <term> "/" | factor

<factor> ::= (<expression>) | <identifier>

<IOStatement> ::= "READ" (<identifier>) | "WRITE" (<identifier>)

<structuredStatement> ::= <cmpdstmt> | <ifStatement> <whileStatement>

<ifStatement> ::= "WHETHER" (<condition>) "EXECUTE" {<statement>}
"OTHERWISE" {<statement>}

<whileStatement> ::= "WHILE" (<condition>) "DO" <statement>

<condition> ::= <expression> <relation> <expression>

<relation> ::= < | <= | == | != |<> | >= | >