```
program ::= metainfo vardeclarationblock executionblock .

metainfo ::= 'PROG' identifier '.' .

vardeclarationblock ::= 'VAR' vardeclarationlist 'END_VAR' .
vardeclarationlist ::= vardeclaration vardeclarationlist | epsilon .
vardeclaration ::= varinout vartype identifier '.' .
varinout ::= epsilon | '?' | '!' | '?' '!' .
vartype ::= 'int' | 'string' | 'float' .

executionblock ::= 'EXEC' statementlist 'END_EXEC' .
statementlist ::= statement statementlist | comment statementlist | epsilon .

comment ::= '/*' word '*/' .

statement ::= statementblock | identifier assignop expr '.' | ifthenelse_stmt |
for_stmt | while_stmt .
statementblock ::= 'begin' statementlist 'end' .

assignop ::= '=' | '+=' | '-=' | '*=' | '/=' .

ifthenelse_stmt ::= 'if' condition 'then' statementlist else_o 'endif' .
else_o ::= epsilon | 'else' statementlist | 'elif' condition 'then'
statementlist else_o .

for_stmt ::= 'for' identifier '=' expr '.' condition '.' identifier assignop
expr 'do' statementlist 'endfor' .
while_stmt ::= 'while' condition 'do' statementlist 'endwhile' .

expr ::= intconst | floatconst | stringconst | identifier | expr '+' expr | expr
'-' expr | expr '*' expr | expr '/' expr | '-' expr | '(' expr ')' .
condition ::= expr '==' expr | expr '~=' expr | expr '<' expr | expr '>' expr |
expr '<=' expr |
                    expr '>=' expr | condition '&&' condition | condition '||'
condition | '~' condition || '(' condition ')' .

intconst ::= digit | digit intconst .
floatconst ::= intconst '.' intconst .
stringconst ::= '"' word '"' .

identifier ::= letter | letter ledis .

ledi ::= letter | digit .
ledis ::= ledi ledis | ledi .

letter ::= 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k' | 'l'
| 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u' | 'v' | 'w' | 'x' | 'y' |
'z' | 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' | 'L' |
'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' |
'Z' .
digit ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' .

word ::= ascii word | epsilon .
ascii ::= all ascii characters, special characters are escaped with backslash
```