```
<program>::=<top_level_statements>
<top_level_statements>::=
  <top_level_statements> <func_decl> |
  <top_level_statements> <statements> |
  EMPTY
<statements>::=
  NEWLINE <statements> |
  <statements> NEWLINE <statement> |
  <statements> NEWLINE |
  <statement>
<statement>::=
  'write' <expression_list> |
  'read to' <identifier> |
  <condition> |
  <while> |
  <foreach> |
  'return' <expression> |
  <assignment> |
  <func_exec> |
  NEWLINE
<condition>::=
  'if' <expression> 'then' NEWLINE INDENT <statements> DEDENT
  ('else  if' <expression> 'then' NEWLINE INDENT <statements> DEDENT)*
  ['else' NEWLINE INDENT <statements> DEDENT]
<while>::='while' <expression> 'do' NEWLINE INDENT <statements> DEDENT
<foreach>::='for each' <identifier> <foreach_list> 'do' NEWLINE INDENT <statements> DEDENT
<assignment>::=
  <identifer> 'equals' <expression> |
  'increase' <identifier> 'by' <expression> |
  'decrease' <identifier> 'by' <expression> |
  'multiply' <identifier> 'by' <expression> |
  'divide' <identifier> 'by' <expression>
<expression>::=
  <expression> 'and' <expression> |
  <expression> 'or' <expression> |
  'not' <expression> |
  <bool> |
  <comparison>
<comparison>
  <comparable> 'is less than' <comparable> |
  <comparable> 'is greater than' <comparable> |
  <comparable> 'is' <comparable> 'or more' |
  <comparable> 'is' <comparable> 'or less' |
  <comparable> 'is between' <comparable> 'and' <comparable> |
  <comparable> 'is' <comparable> |
  <aritm_expr>
<aritm_expr>::=<aritm_expr> ('plus' | 'minus') <term> | <term>
<term>::=<term> ('times' | 'divided by') <factor> | <factor>
<factor>::=
  <factor> 'modulo' <factor> |
  '('<expression>')' |
  FLOAT |
  INTEGER |
  <func_exec> |
  <index> 'of' <indexable>
  <identifier>
  <string>
  <array>
<func_decl>::=<identifier> ['with' <identifier_list>] 'does' NEWLINE INDENT <statements> DEDENT
<func_exec>::='do' <identifier> ['with' <expression_list>]

<identifier_list>::=[<identifier_list> ','] <identifier>
<expresssion_list>::=[<expression_list> ','] <expression>
<foreach_list>::=
  'in' <expression> |
  'from' <foreach_elem> 'to' <foreach_elem>
<foreach_elem>::=<identifier> | INTEGER
<comparable>::=<aritm_expr> | <string> | <array>
<index>::=
  /^\d*(11|12|13)th$/ |
  /^\d*1st$/ |
  /^\d*2nd$/ |
  /^\d*3rd$/ |
  /^\d+th$/ |
  /^<identifier>th$/ |
  'last'
<identifier>::=/^[a-zA-Z]+$/
<string>::=/".*?"/
<array>::='[' [<expression_list>] ']'
<bool>::= 'true' | 'false'
<comment>::=/#.*?$/
```