Grammatik

```
<program>::= <stmt_list>

<stmt_list>::=  <stmt><stmt_list>
              |  <stmt>

<stmt>::=  <return_stmt>
          |   <io_stmt>
          |   <sel_stmt>
          |   <iter_stmt>
          |   <data_stmt>
          |   <assign_stmt>
          |   <function_stmt>
          |   <expr>

<function_stmt::= <function_def> | <function_call>

<function_def>::= FUNCTION <func_name> '('<parameter_list>*')'
                  <stmt_list> /FUNCTION

<function_call>::= <func_name> '('<argument_list>*')'

<argument_list>::=  <stmt>
                   | <stmt>, <argument_list>

<parameter_list>::= '<' <type> '>' <var_dec>
                   | '<' <type> '>' <var_dec>, <parameter_list>

<comments>::= # comments /#

<sel_stmt>::=   IF '(' <expr> ')'  <stmt_list> ELSE <stmt_list> /IF
      |   IF '(' <expr> ')'  <stmt_list> ELSE <elseif> /IF
      |   IF '(' <expr> ')'  <stmt_list> /IF
<elseif>::=  IF '(' <expr> ')` <stmt_list>  ELSE <elseif>
      |   IF '(' <expr> ')` <stmt_list>  ELSE <stmt_list>
      |   IF '(' <expr> ')' <stmt_list>

<iter_stmt>::= WHILE '(' <expr> ')'<stmt_list>/WHILE
      |   FOR '('<iter_var> IN <num> TO <num> INCBY <num>  ')' <stmt_list> /FOR
      |   EACH '('<iter_var> IN <var_dec> ')' <stmt_list> /EACH

<iter_var>::= <' <type> '>' <var_dec>

<data_stmt>::= EMPTY <var_dec>
               | REMOVE <var_dec> '[' <NUM> ']'
               | REMOVE <var_dec> '{' <STR> '}'
               | ADD <var_dec> <list>
               | ADD <var_dec> : <hash>

<list>::= '[' <type_list> ']' | '['  ']'
<hash>::='{' <type_hash> '}' | {'  '}'

<type_list>::= <type> | <type_list> ',' <type>
<type_hash>::= <str> '>' <type> | <type_hash> ','<str> '>' <type>

<assign_stmt>::=   DECL '<' <type> '>' <var_dec>: /DECL
                    |  DECL '<' <type> '>' <var_dec>: <expr> /DECL
                    |  HASH <var_dec> : <hash> /HASH
                    |  LIST '<' <type>'>' <var_dec>: <list> /LIST


<io_stmt>::=   <print_stmt>
            | <read_stmt>
```

```
<print_stmt>::= PRINT <stmt_list> /PRINT

<read_stmt>::= READ <var_dec> /READ | READ '<' <type> '>' <var_dec> /READ

<return_stmt>::= DONE  <stmt_list> /DONE

<expr>::= <expr> <operator_a> <term>
          | <term>

<term>::= <term> <operator_b> <log>
          | <log>

<log>::= <log> <log_operator> <comp>
         | <log_operator_not> <comp>
         | <comp>

<comp>::= <comp> <comp_operator> <factor>
        | <factor>

<factor>::= <function_call>
             | (<expr>)
             | <type>
             | <var_dec>
             | <data>
             | <list>
             | <hash>

<data>::= <var_dec> '[' <num> ']`    // call list
          | <var_dec> '[' <var_dec> ']`    // call list
          | <var_dec> '{' <str> '}`    // call hash
          | <var_dec> '{' <var_dec> '}`    // call hash

<comp_operator>::= == | =/= | > | >= | < | <=

<log_operator>::= && | '||' | AND | OR
<log_operator_not>::= ! | NOT

<operator_a>::=  + | -
<operator_b>::=  * | / |%

<type>::= <num>
           |  <str>
           |  <bool>
           |  <all>

<num>::=   Float
      |  '-' Float

<str>::= sequence /"[\w\s!\?]*"/

<var_dec>::= sequence /[A-Z][a_z0-9_]*/

<func_name>::= sequence /[A-Z_]+/

<bool>::= TRUE | FALSE
```