

Special symbols:

-operators: + - * / % <- LESS_THAN GREATER_THAN LESS_OR_EQUAL_THAN
GREATER_OR_EQUAL_THAN EQUAL IS BINARY_AND BINARY_OR
-separators: ; space @ [] () ""
-reserved words:
CREATE_PROGRAM DESTROY_PROGRAM SPAWN nr flag char IN OUT CHECK LOOP =>
AND OR

Identifiers:

identifier = letter | letter {letter} {digit}
letter = "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"
digit = "0" | "1" | ... | "9"

Constants:

1. nr
nr = ("+" | "-") digit {digit}
2. flag
flag = "true" | "false" | "0" | "1"
3. char
char = letter | digit

+
-
*
/
%
<-
;
space
@
[
]
(
)
"
LESS_THAN
GREATER_THAN
LESS_OR_EQUAL_THAN
GREATER_OR_EQUAL_THAN
EQUAL
IS
BINARY_AND
BINARY_OR
CREATE_PROGRAM
DESTROY_PROGRAM
SPAWN
nr
flag
char
IN
OUT
CHECK
LOOP
=>
AND
OR

```

//program
program = "CREATE_PROGRAM" ";" decllist stmtlist "DESTROY_PROGRAM" ";"

//types
typesimple = "nr" | "flag" | "char"
arraydecl = typesimple "[" nr "]"
type = typesimple | arraydecl

//declarations
declaration = "SPAWN" type IDENTIFIER ";"
decllist = declaration | declaration decllist

//expressions
expression = arithmexp | binaryexp | IDENTIFIER
arithmexp = arithmexp ("+" | "-" | "*" | "/" | "%") IDENTIFIER | IDENTIFIER
binaryexp = binaryexp ("BINARY_AND" | "BINARY_OR") IDENTIFIER | IDENTIFIER

//statements
stmt = simplstmt | structstmt
stmtlist = stmt | stmt stmtlist
simplstmt = (assignstmt | iostmt) ";"
cmpdstmt = "@" stmtlist "@"
structstmt = cmpdstmt | ifstmt | whilestmt
assignstmt = IDENTIFIER "<-" expression
iostmt = ("IN" | "OUT") "" IDENTIFIER ""
ifstmt = "CHECK" condition "=>" stmt
whilestmt = "LOOP" condition "=>" stmt

//conditionals
condition = expression RELATION expression
RELATION = "LESS_THAN" | "GREATER_THAN" | "LESS_OR_EQUAL_THAN" |
"GREATER_OR_EQUAL_THAN" | "EQUAL" | "IS"

```