https://github.com/IanisT227/FLCD_Ianis_Teja

Lexic.txt

Alphabet:

a. Upper (A-Z) and lower case letters(a-z) of the english alphabet
b. Tilda character “~”;
c. Decimal digits(0-9);

Lexic:

a.Special symbols, representing:

-operators + - * / < <= = >= isNot and

-separators : () [] space {}

-reserved words:

var read write whileTrue endWhileTrue checkIf Integer String List

b.identifiers

letter::= A | B | C |... |Z | a | b | c | … | z |

identifier::= letter | letter{letter}

c.constants

1. **Integer - rule:**

digit::= 0 | 1 | … | 9

nonZeroDigit::= 1 | 2 | … | 9

sign::= - | +

constNumber::= digit | nonZeroDigit{digit} | sign nonZeroDigit {digit}

2. **String - rule:**

constString::= “string”

string::= char{string}

char::= letter|digit

3. **List - Rule**

list::= string {string} | number {number}

Syntax.in

Program starts and ends with ~

Syntactic rules:

declaration::= var <identifier> : <type>

type::= Integer|String

arraydecl::= var <identifier> : List(<type>, <number>)

stmt ::= simplstmt | structstmt

simplstmt ::= assignstmt | printstmt

assignstmt ::= “read” “(“ identifier “,” (identifier | constant) “)”

expression ::= expression "+" term | term

term ::= term ["*”|"/”] factor | factor

factor ::= "(" expression ")" | IDENTIFIER

printstmt ::= "write" "(" identifier")" | “write” “(“ constant “)”

structstmt ::= stmtlist | ifstmt | whilestmt

ifstmt ::= "checkIf(" condition ")" stmt ["else" stmt]

whilestmt ::= "whileTrue(" condition ")" stmt “endWhileTrue”

condition ::= expression RELATION expression

stmtlist ::= stmt | stmt “;” stmtlist

RELATION ::= "<" | "<=" | "=" | ">=" | ">"