FLCD – lab2

Minilanguage specification

Github: https://github.com/IuliaPapureanu/FLCD

Specification (file Lexic.txt)

Alphabet:

a. Upper (A-Z) and lower case letters (a-z) of the English alphabet

b. Underline character '_';

c. Decimal digits (0-9);

1. Lexic:

a.Special symbols, representing:

- operators + - * / < <= = >= == ^ % =

- separators [ ] { } : ; space

- reserved words:

if then else endif while endwhile read print int string

array var

b.identifiers

-a sequence of letters and digits, such that the first character is a letter; the rule is:

identifier = letter{(letter|digit)}

letter = "A" | "B" | . ..| "Z"|”a”|”b”|...|”z”

digit = "0" | "1" |...| "9"

c.constants

1.integer - rule:

noconst:=”0”| [(”+”|”-”)] no

no:= nonzero[{digit}]

nonzero := “1”|”2”|...|”9”

2.string

conststring:="string"

string:={char}

char:=letter|digit

2. Syntax:

The words - predefined tokens are specified between " and ":

Sintactical rules: (file Syntax.in)

type1 = "string" | "int"

arraydecl = type1 “array” "[" nr "]"

type = type1|arraydecl

declaration =type IDENTIFIER

decllist = declaration | declaration ";" decllist

stmtlist = stmt | stmt ";" stmtlist

stmt = simplstmt

simplstmt = assignstmt | iostmt

assignstmt = identifier "=" expression

expression = expression "+" term | term

term = term "*" factor | factor

factor = "(" expression ")" | identifier

iostmt = ("read" | "print") "(" identifier ")"

ifstmt = "if" “(” condition “)” "then" stmt ["else" stmt] “endif”

whilestmt = "while" “(” condition “)” stmt “endwhile”

condition = expression RELATION expression

RELATION = "<" | "<=" | "==" | "<>" | ">=" | ">"|”!=”