

LEXIC

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 + - * div mod = is > >= < <= and or not .
- separators : ; space []
- reserved words cst verify yes no print var read
return list go as with

- b. Identifiers

identifier = letter {“_” | letter | digit}

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

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

- c. Constants

- 1. Integer

int = [“+” | “-”] nr | “0”

nr = ndigit {digit}

ndigit = "1" | "2" | ... | "9"

- 2. Character

char = “'” (letter | digit) “'”

- 3. String

str = “ ” char {char} “ ”

- 4. Boolean

bool = “T” | “F”

TOKEN

+

-

*

div

mod

=

is

>

>=

<

<=

and

or

not

.

:

;

space

[

]

words

cst

verify

yes

no

print

var

read

return

list

go

as

with

T

F

SYNTAX

program = {declist | compstmt}

declist = declaration {declaration}

declaration = type identifier “;”

type = “var” | “cst” | listdcl

listdcl = “list” “[nr ”]

compstmt = { stmtlist }

stmtlist = stmt {stmt}

stmt = simplestmt | structstmt

simplestmt = assignstmt | iostmt

assignstmt = identifier “=” expr “;”

expr = term { (“+” | “-”) expr }

term = factor { (“*” | “div” | “mod”) term }

factor = identifier | int

iostmt = (“read” | “print”) identifier “;”

structstmt = compstmt | ifstmt | forstmt

ifstmt = “verify” cond “yes” “:” stmt [“no” “:” stmt]

cond = expr rel expr

rel = “is” | “<” | “<=” | “>” | “>=”

forstmt = “go” assignstmt “as” cond “with” assignstmt “:” stmt