Laborator 1 LFTC

**1.Specificarea mini limbajului**

* Definirea limbajului

1. Alfabetul
2. Litere ale alfabetului englez(litere mari si mici A-Z a-z) denumite pe scurt litera
3. Cifre zecimale

nonzeroDigit (‘1’-’9’)

zero (‘0’)

1. Caracterele(‘\_’, ‘#’)

2. Lexic

1. Simboluri speciale

* Operatorii +, -, \*, /,=, ==, <, <=, >, >=, %
* Separatori [], {}, spatiu, ;, :
* Cuvinte rezervate if, else, then, for, while, const, string, integer, char, boolean, hello, bye, read, write, list, of, do, GO, DONE

b. Identificatori

- secventa de litere, cifre și caractere care începe cu litera si contine cel putin o litera urmata de litere sau cifre

Regula:

identificator::=litera{(litera|cifra)}

litera ::= “A”|”B”|...|”Z”|”a”|”b”|...|”z”

cifra ::= “0” | “1” | …|”9”

c. Constante

* De tip intreg

const::=(+|-)nonzeroNumber | number

nonzeroNumber::= nonzeroDigit{(nonzeroDigit|zero)}

number ::= zero | {nonzeroNumber}

positiveNumber0::=+nonzeroNumber|zero

* De tip caracter

const=::’litera’|’nonzeroDigit‘|‘zero’

* De tip string

const=::”string”

string::=caracter{string}

caracter::=litera|nonzeroDigit|zero

d.Relatii

relatie::= ”<”|”<=”|”>”|”>=”|”==”|”<>”

* Sintaxa

Reguli sintactice

program::=”hello” listaDeclaratii”;”instructiuneCompusa “bye”

listaDeclaratii ::= declaratie | declaratie”;”listaDeclaratii

declaratie::= identificator “:” tip

declaratieLista::=”list”[number] “of” tip

tip::=”string”|“integer”|“char”|“boolean”

tip2::=tip | declaratieLista

instructiuneCompusa::=”GO” listaInstructiuni “DONE”

listaInstructiuni::=instructiune | instructiune listaInstructiuni

instructiune::=instructiuneSimpla | instructiuneComplexa

instructiuneSimpla::=instructiuneAtribuire | instructiuneInputOutput

instructiuneAtribuire::=identificator“=”expresie”;”|identificator”=”identificator”;”|

identificator”=” ”elementLista” ”;”

expresie::=expresie operatie termen

operatie::=”+”|”-”|”/”|”%”|”\*”

termen::=termen operatie factor | factor

factor::=”(“expresie”)” | identificator | numar

elementLista::= lista”[“index”]”

index::=positiveNumber0

instructiuneInputOutput ::= “read” “(“identificator”)””;” | “write” “(“identificator”)”;”

instructiuneComplexa::=instructiuneIf | instructiuneFor | instructiuneWhile

instructiuneIf::=”if” conditie “then” “{“ listaInstructiuni “}” [“else” ”{“ listaInstructiuni “}”]

instructiuneWhile::=”while” conditie “do” “{“ listaInstructiuni “}”

intructiuneFor::=”for” elem “in” colectie “do” ”{“ listaInstructiuni “}”

conditie::=expresie relatie expresie

colectie::=identificator

**2.Exemple de programe scrise cu ajutorul mini limbajului**

1.Verificare daca un numar intreg este multiplu de 3

hello

numar:integer; rest:integer; result:string;

GO

read(numar);

rest=numar%3;

if rest==0 then {result=”multiplu3”}

else{result=”nuEsteMultiplu3”}

write(result);

DONE

bye

2. Determinarea maximului dintr-un vector

hello

numere[4] of integer; max:integer; elem:integer

GO

max=numere[0];

for elem in numere do

{

if elem > max then

{

max=elem;

}

}

write(max);

DONE

bye