## Exemplu de specificare minilimbaj de programare

**1 .Definirea limbajului:**

1.1 Alfabetul

a. Literele mari si mici ale alfabetului englez (A-Z si a-z)

b. Caracterele \_ + - \* / : = < > [ ] { } : " '

c. Cifrele zecimale (0-9);

**Lexic**

* Simboluri speciale, reprezentand:

- **operatorii** + - \* / := < <= = >=

- **separatorii** [ ] { } : ; spatiu " '

- **cuvinte rezervate:** ARRAY CHAR INTGER IF ELSE READ THEN VAR WHILE BEGIN END WRITE

- **RELATIE** >= <= > < =

**Identificatori:** o secventa de litere si cifre, care incepe cu o litera; regula este:

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

litera ::= "A" | "B" | . ..| "Z" | "a" | "b" | . ..| "z"

cifra ::= "0" | "1" |...| "9"

cifra\_nenula ::= "1" | "2" |...| "9"

* **constante**

1.**intregi** - regula:

const ::= [semn]cifra\_nenula{numar} | "0"

numar ::= cifra{numar}

semn ::= "+" | "-"

2.**naturale** - regula:

numar­\_natural ::= cifra\_nenula{cifra} | "0"

3.**caracter**

Character ::= 'litera'|'cifra'|' '

4.**string**

ConstCaracter ::= "string"

string ::= caracter{string}

caracter:=litera|cifra|' '

**2. Sintaxa:**

-------

Cuvinte - atomii predefiniti sunt scrisi intre ghilimele( " si "):

**a) Reguli sintactice:**

------------------

program ::= "VAR" listaDeclaratii ";" instructiuneCompusa "."

listaDeclaratii ::= declaratie | declaratie ";" listaDeclaratii

declaratie ::= IDENTIFICATOR ":" tip

tip1 ::= "CHAR" | "INTEGER"

tip2 ::= "ARRAY" tip1 "[ "dimensiune "]"

tip ::= tip1|tip2

dimensiune ::= IDENTIFICATOR | numar­\_natural

accesPozitie ::= IDENTIFICATOR | numar­\_natural

accesVector ::= IDENTIFICATOR "[ "accesPozitie"]"

instructiuneCompusa ::= "BEGIN" listaInstructiuni "END"

listaInstructiuni ::= instructiune ";" | instructiune ";" listaInstructiuni

instructiune ::= instructiuneSimpla | instructiuneStructurata

instructiuneSimpla ::= atribuire | io

atribuire ::= IDENTIFICATOR "=" expresie

expresie ::= expresie ( "+" | "-" ) termen | termen

termen ::= termen ( "\*" | "/") factor | factor

factor ::= "(" expresie ")" | IDENTIFICATOR

io ::= "READ" "(" IDENTIFICATOR ")"| "WRITE" "(" IDENTIFICATOR ")"

instructiuneStructurata ::= instructiuneCompusa | instructiuneIf | instructiuneWhile

instructiuneIf ::= "IF" conditie "THEN" instructiune ["ELSE" instructiune]

instructiuneWhile ::= "WHILE" conditie "DO" instructiune

conditie ::= expresie RELATIE expresie

**b) Reguli lexicale:**

----------------

IDENTIFICATOR ::= litera| litera{litera}{cifra}

litera ::= "A" | "B" | . ..| "Z"

cifra ::= "0" | "1" |...| "9"

RELATIE ::= "<" | "<=" | "=" | "<>" | ">=" | ">"

Atomii sunt codificati conform urmatoarei tabele:

- identificatori - cod 0

- constante - cod 1

- cuvinte rezervate: fiecare cuvant are propriul cod

- separatori: fiecare separator are propriul cod

**Codificare**:

|  |  |
| --- | --- |
| **Tip atom** | **cod** |
| identificator | 0 |
| constanta | 1 |
| program | 2 |
| array | 3 |
| var | 4 |
| integer | 5 |
| real | 6 |
| boolean | 7 |
| begin | 8 |
| end | 9 |
| read | 10 |
| write | 11 |
| do | 12 |
| if | 13 |
| then | 14 |
| else | 15 |
| : | 16 |
| ; | 17 |
| , | 18 |
| . | 19 |
| + | 20 |
| \* | 21 |
| - | 22 |
| / | 23 |
| ( | 24 |
| ) | 25 |
| [ | 26 |
| ] | 27 |
| < | 28 |
| > | 29 |
| = | 30 |
| <= | 31 |
| >= | 32 |

**Exemplu de programe**

VAR n:INTEGER;

d:INTEGER;

sum:INTEGER;

nr:INTEGER;

BEGIN

sum=0;

READ(n);

WHILE n >1

DO

BEGIN

READ(nr);

sum=sum+nr;

n=n-1;

END

END

.

2)

VAR A:INTEGER;

B:INTEGER;

BEGIN

READ(A);

IF A>B

WRITE(A);

ELSE

WRITE(B);

END

.