**Specificarea minilimbajului de programare**

**- LFTC – Laboratorul 1 –**

Definirea Limbajului:

1. Alfabetul limbajului:

* Litere mici (a-z) ale alfabetului englez
* Litere mari (A-Z) ale alfabetului englez
* Cifrele sistemului zecimal (0-9)
* Caracterul subliniere ‘\_’

1. Lexic:
2. Simboluri speciale pentru reprezentarea:

- Operatorilor: + - \* / = > < >= <= == !=

- Delimitatori: ( ) { } [ ] ;

- Cuvinte rezervate: daca, altfel, cat\_timp, executa, int, char, lista, citeste, scrie

b) Identificatori – o succesiune de litere | litere + cifre, primul caracter fiind obligatoriu litera

Regula de constructie:

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

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

- cifra := “0” | “1” | … | “9”

c) Constante

1. Intregi

- constnr := nr | +nr | -nr

- nr := cifra{nr}

2. caracter

- constcaracter := ‘litera’ | ‘cifra’

3. sir de caractere

- constsir := “sirdecaractere”

- sirdecaractere := constcaracter{sirdecaractere}

Sintaxa Limbajului:

Cuvintele: atomii lexicali rpedefiniti sunt scrisi intre ghilimele

1. Reguli Sintactice:

program := “begin” listadeclaratii listainstructiuni “end”

listadeclaratii:= declaratie | declaratie “;”

declaratie := tip identificator “;”

tip := tipsimplu | tipcompus

tipsimplu := “int” | “char” | “real”

tipcompus := “lista” tipsimplu”[“dimensiune”]”

dimensiune := constnr

listainstructiuni := instructiune “;”| instructiune “;” {listainstructiuni}

instructiune := instructiunesimpla | instructiunecompusa

instructiunesimpla := atribuire | instructiuneIO

atribuire := identificator “=” expresie

expresie := expresie (“+”|”-“) termen | termen

termen := termen (“\*”|”/”) factor | factor

factor := “(“ expresie “)” | identificator | constnr | constcaracter

instructiuneIO := (“citeste” | “scrie”) identificator | “scrie” constsir

instructiunecompusa := instrdaca | instrcat\_timp | instrex\_cat\_timp

instrdaca:= “daca” “(“ conditie “)” “{“ listainstructiuni “}” [ “altfel” “{“ listainstructiuni “}” ]

conditie := expresie relatie expresie

instrcat\_timp := “cat\_timp” “(“ conditie “)” “{“ listainstructiuni “}”

instrex\_cat\_timo := “executa” “{“ listainstructiuni “}” “cat\_timp” “(“ conditie “)” “;”

1. Reguli lexicale:

Identificator := litera | liteara{litera}{cifra}

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

cifra := “0” | “1” | … | “9”

relatie := “<” | “>” | “<=” | “>=” | “==” | “!=”

**Codificare:**

* Identificatori: cod 0
* Constante : cod 1
* Cuvinte cheie, operatori, separatori : cod propriu

|  |  |  |  |
| --- | --- | --- | --- |
| **Tip atom** | **Cod** | **Tip atom** | **Cod** |
| identificator | 0 | ; | 27 |
| constanta | 1 | + | 28 |
| Begin | 2 | - | 29 |
| End | 3 | \* | 30 |
| int | 4 | / | 31 |
| real | 5 | . | 32 |
| char | 6 |  |  |
| lista | 7 |  |  |
| citeste | 8 |  |  |
| scrie | 9 |  |  |
| daca | 10 |  |  |
| altfel | 11 |  |  |
| cattimp | 12 |  |  |
| executa | 13 |  |  |
| = | 14 |  |  |
| ( | 15 |  |  |
| ) | 16 |  |  |
| [ | 17 |  |  |
| ] | 18 |  |  |
| { | 19 |  |  |
| } | 20 |  |  |
| == | 21 |  |  |
| != | 22 |  |  |
| < | 23 |  |  |
| > | 24 |  |  |
| <= | 25 |  |  |
| >= | 26 |  |  |