

Lucrarea 2

Funcții predefinite

Funcția de intrare

Funcția de intrare standard în Lisp este funcția **READ**.

Dacă funcția nu are nici un argument, citirea se face de la tastatură.

Exemplu:

```
> (read)
```

```
3456
```

```
3456
```

```
> (read)
```

```
"cati ani ai"
```

```
"cati ani ai"
```

După apelul funcției se așteaptă citirea pe o durată nelimitată, până la introducerea de la tastatură, validarea citirii se face cu ENTER, apoi se returnează valoarea citită.

Citirea în Lisp nu presupune numai citirea caracterelor și returnarea lor ca șir de caractere, ci Lisp-ul analizează ceea ce citește și returnează obiectul Lisp rezultat.

Funcții predefinite

Funcția de ieșire

Cea mai generală **funcție de ieșire** în Lisp este funcția **FORMAT** .

Ea are două sau mai multe argumente:

primul indică unde va fi tipărit rezultatul, al doilea este un șablon și de caractere, iar restul argumentelor sunt de obicei obiecte ale căror reprezentări tipărite sunt inserate în șablon.

Exemplu:

```
> (format t "~A plus ~A egal ~A.~%" 2 3 (+ 2 3))
```

```
2 plus 3 egal 5.
```

```
NIL
```

Funcția returnează pe prima linie ceea ce afișează FORMAT, iar pe a doua linie se afișează valoarea returnată de apelul Format. Atunci când funcția este utilizată în interiorul programelor, valoarea de pe a doua linie nu este vizibilă.

Primul argument al funcției -t, indică faptul ca afișarea va fi făcută în locul implicit.

Al doilea argument este un șir care servește drept șablon pentru afișare. În cadrul acestui șir, fiecare caracter ~A indică o poziție care va fi ocupată, iar ~% - indică o linie nouă. Pozițiile vor fi ocupate de valorile argumentelor următoare în ordinea în care ele apar în Format.

Funcții predefinite

Funcții aritmetice

Lispul are o bibliotecă foarte mare de funcții aritmetice. Vom prezenta numai câteva dintre acestea:

+ adunare: se pot aduna oricâte argumente

> (+ 2 3 4 5)

14

- scădere: dacă avem un singur argument, atunci se obține numărul negativ:

> (- 7)

-7

> (- 8 3 2)

3

*** înmulțire:**

> (* 3 5 1)

15

/ împărțire:

> (/ 6 3)

2

> (/ 75 3 5)

5

Funcții predefinite

Dacă argumentele funcțiilor aritmetice sunt întregi, rezultatul va fi tot întreg.

Dacă unul din argumente este real, atunci rezultatul va fi real.

`> (+ 6 (- 7.5 3))`

10.5

Excepție este dacă împărțim două numere întregi care nu dau un rezultat exact, atunci rezultatul este un număr real.

`> (/ 25 4 3)`

2.08333

Funcții aritmetice

Toate operațiile de mai sus se aplică și argumentelor numere complexe:

`> (+ #c(1 2) #c(3 2))`

`#C(4 4)`

`> (- #c(1 2) #c(3 2))`

`-2`

`> (* #c(1 2) #c(3 2))`

`#C(-1 8)`

`> (/ #c(1 2) #c(3 2))`

`#C(0.538462 0.307692)`

Funcții predefinite

Funcții aritmetice

Funcțiile ***floor*** și ***ceiling*** , rotunjesc o valoare în jos, respectiv în sus.

> (floor 2)

2

> (floor 4.66)

4

> (floor 2.33)

2

> (ceiling 2.99)

3

> (ceiling 2.25)

3

> (floor (* 2 (/ 5 3)))

3

> (ceiling (* 2 (/ 5 3)))

4

Funcții predefinite

Pentru numerele întregi, există două funcții care returnează restul Împărțirii celor două numere:

Funcția **REM** sau **MOD**. Ambele au același efect.

> (rem 5 3)

2

> (mod 14 5)

4

Există două forme LISP de **incrementare**, respectiv **decrementare**, fără efecte laterale:

Funcțiile **1+** respectiv **1-**

> (1+ 5)

6

> (1- 8)

7

Funcția de determinare a valorii absolute: **ABS**

> (abs 5)

5

> (abs -3)

3

Dacă argumentul este număr complex, rezultatul este modulul:

> (abs #c(3 -4))

5

Funcții aritmetice

Funcții predefinite

Funcția radical : SQRT - are un singur argument

> (sqrt 3)

1.73205

> (sqrt 5 4)

error: too many arguments

Funcții aritmetice

Funcția ridicare la putere EXPT – are 2 argumente, primul este baza, iar al doilea este puterea.

>(EXPT 2 3)

8

Funcțiile MIN si MAX –calculează minimul și maximul unui șir de n numere:

>(MIN 5 2 -3 7 8)

-3

>(MAX 2 1 8 3 7)

8

Probleme

1. Transpuneți în LISP următoarele expresii și apoi evaluați-le:
 $(25 + 30) * 15/2$
 $6 * 3.1416$
2. Determinați câtu întreg pentru media aritmetică a numerelor:
5, 6.7, -23.2, 75 și 100.3
3. Determinați restul împărțirii numerelor:
365 și 12, 13 și 467
4. Scrieți sub formă de expresie LISP soluțiile următoarei ecuații de gradul 2, apoi evaluați expresiile:
 $2x^2 + 7x + 5 = 0$
5. Determinați rezultatul întreg prin rotunjire în jos, respectiv în sus, apoi incrementați , respectiv decrementați rezultatul pentru următoarea expresie, :
 $(5 * 2.25 - 7.13) / (45 - 25 / 5)$
6. Evaluați următoarele forme:
 $(* (MAX 3 4 5) (MIN 3 4 5))$
 $(EXPT (MAX 3 1 4) (MAX 2 7 1))$
 $(REM (+ 5 7 13) (- 14 1))$
7. Afișați rezultatul de la problema 3 sub forma unui mesaj:
Ex: restul împărțirii numerelor ... și ... este:...
8. Afișați un mesaj înainte de citirea unei variabile, apoi citiți variabila.
Ex: "a =" citire valoare