

LIMBAJUL DE PROGRAMARE PROLOG

Prolog provine de la **Pro**gramming in **Log**ic. Prolog este un limbaj de programare logică bazat pe principiile logicii. Pentru limbajele de programare se pot distinge mai multe categorii, precum:

- Limbaje structurate (C, C++, Java, etc.)
- Limbaje logice (Prolog, Lisp, etc.)

Limbajul Prolog conține o bază de cunoștințe pentru care se stabilesc anumite relații. În limbaj **nu se întâlnesc noțiunile de procedură sau de atribuire**; variabilele de program primesc valori prin stabilirea anumitor legături. Execuția unui program Prolog presupune stabilirea unei cerințe sau a unui scop care se verifică în funcție de baza de cunoștințe. Variabile de program sunt identificatori care încep cu litere mari. Pentru variabilele de program se vor asocia valori printr-un proces de tip “matching” îmbinat cu unul de tip backtracking recursiv.

TIPURI DE OBIECTE

Program Prolog - reprezintă o descriere de obiecte și a relațiilor dintre obiecte.

Tipuri de date:

1. TIP SIMPLU

A. Tipuri de date elementare (integer, real, char, string, **symbol**)

Observație: Datele de tip **symbol** reprezintă șiruri de caractere care încep cu literă mică.

B. Tipuri de date definite de utilizator

2. TIP COMPLEX

a. LISTA

b. COMPUS (listă de liste)

SECȚIUNILE UNUI PROGRAM PROLOG

1. **constants** //definirea constantelor

2. **domains** // se definesc domeniile obiectelor din program

domeniu1=symbol

domeniu2=integer* // se definește domeniu2 ca fiind un domeniu de tip listă de numere întregi

3. **predicates** // se stabilesc legăturile dintre obiectele din program

nume_predicat(arg1, arg2, ..., argn)

OBS: *arg1, arg2, ..., argn* reprezintă domeniile obiectelor aflate în relația precizată

4. **clauses** // se definesc în mod explicit predicatele din secțiunea anterioară
nume_clauză(arg1, ..., argn).

$c_{n+1}(.) :- c_1(.), c_2(.), \dots, c_n(.)$.

Obs:

Clauza $c_{n+1}(.)$ = Adevărat dacă $c_1(.)$ SI LOGIC $c_2(.)$ SI LOGIC ... SI LOGIC $c_n(.)$ = Adevărat

5. goal // se stabilesc scopurile, cerințele de program

Observații:

1. Tipuri de clauze:
 - a. Clauze reguli (nume_clauza(.) :- c1, c2, ..., cn.)
 - b. Clauze fapt (nume_clauza(lista_argumente).)
2. Orice clauză se încheie cu punct.
3. Elementele logice sunt prezente în program prin:
 - “,” desemnează ȘI LOGIC
 - “;” desemnează SAU LOGIC
 - “:-” desemnează IMPLICAȚIA LOGICĂ (DACĂ)
4. Operatori în Prolog:
 - a. Operatorii aritmetici: +, -, *, /, mod (restul împărțirii nr. întregi)
 - b. Operatori relaționali: >, >=, <, <=, <>, !=
 - c. Negația logică: NOT

APLICAȚIA 1.

Se consideră următoarea bază de cunoștințe cu informații privind anumite firme și produsele distribuite.

- a. Magic srl vinde calculatoare și accesorii.
- b. Alfa sa vinde televizoare și calculatoare.
- c. Beta sa vinde accesorii.

CERINȚE

- A. Vinde Alfa SA calculatoare?
- B. Vinde BetaSA televizoare?
- C. Ce vinde Magic SRL?
- D. Cine vinde calculatoare?
- E. Vinde Alfa SA ceva?
- F. Vinde cineva accesorii?
- G. Cine vinde la fel ca BetaSA un anumit produs?
- H. Cine vinde ce?
- I. Vinde cineva ceva?

PROGRAM PROLOG

domains

nume_firma=symbol
nume_produș=symbol

predicates

vinde(nume_firma, nume_produș)

clauses

vinde(magicsrl,calculatoare).
vinde(magicsrl,accesorii).
vinde(alfasa,calculatoare).
vinde(alfasa,televizoare).
vinde(betasa,accesorii).

Executie: ALT+R (Run)

DIALOG (GOAL)

A. Vinde Alfa SA calculatoare?

vinde(alfasa,calculatoare)

B. Vinde BetaSA televizoare?

vinde(betasa, televizoare)

C. Ce vinde Magic SRL?

vinde(magicsrl, X)

D. Cine vinde calculatoare?

vinde(Y,calculatoare)

E. Vinde Alfa SA ceva?

vinde(alfasa, _)

F. Vinde cineva accesorii?

vinde(_, accesorii)

G. Cine vinde la fel ca BetaSA un anumit produs?

vinde(betasa,X), vinde(Y,X), Y<>"betasa"

H. Cine vinde ce?

vinde(X,Y)

I. Vinde cineva ceva?

vinde(_,_)

APLICAȚIA 2

Se consideră următoarea bază de cunoștințe cu informații privind anumite persoane și produsele distribuite.

- a. Alex vinde trandafiri, lalele și crizanteme.
- b. Ana vinde lalele și crini.
- c. Andu vinde lalele și nuci.

CERINȚE

- 1. Vinde Andrei lalele?
- 2. Cine vinde crizanteme?
- 3. Ce vinde Ana?

4. Vinde cineva crini?
 5. Vinde Andu ceva?
 6. Cine vinde la fel ca Alex un anumit produs?
 7. Ce vinde Andu și nu este floare?
- vinde(andu,X), not(floare(X)).

PRELUCRĂRI ARITMETICE

APLICAȚIA 1: Să se calculeze media aritmetică a două valori reale.

Execuția programului: media_aritmetica(40, 34, Media)

Program PROLOG

```
predicates
    media_aritmetica(real, real, real)
clauses
    media_aritmetica(A, B, M_aritmetica):- M_aritmetica=(A+B)/2.
```

APLICAȚIA 1': Să se calculeze media aritmetică a două valori reale, respectiv, media armonica.

Execuția programului:

⇒ media_aritmetica(40, 34, Media); media_armonica(40, 34, M)

Program PROLOG

```
predicates
    media_aritmetica(real, real, real)
    media_armonica(real, real, real)
clauses
    media_aritmetica(A, B, M_aritmetica):- M_aritmetica=(A+B)/2.
    media_armonica(A, B, M):- M=2/(1/A+1/B), A<>0, B<>0.
```

APLICAȚIA 2: Să se determine maximul/minimul a două valori reale.

Execuția programului: maxim(40, 34, Max); minim(40, 34, Min)

Program PROLOG

```
predicates
    maxim(real, real, real)
    minim(real, real, real)
clauses
    maxim(A,B,A):- A>=B.
    maxim(A,B,B):- A<B.
    minim(A,B,B):- A>=B.
    minim(A,B,A):- A<B.
```

GOAL: maxim(40, 34, Max) minim(21, -4, Min)

APLICAȚIA 3: Să se determine maximul/minimul a trei valori reale.

Program PROLOG

```
predicates
    maxim(real, real, real)
    minim(real, real, real)
    maxim3(real, real, real, real)
    minim3(real, real, real, real)
clauses
    maxim(A,B,A):- A>=B.
    maxim(A,B,B):- A<B.
    minim(A,B,B):- A>=B.
    minim(A,B,A):- A<B.
    maxim3(A,B,C, Max3):- maxim(A,B, R), maxim(R,C, Max3).
    minim3(A,B,C, Min3):- minim(A,B, R), minim(R,C, Min3).
```

GOAL: maxim3(40, 34, 12, Max) minim3(21, -4, 2, Min)

**APLICAȚIA 4: Să se determine maximul/minimul a două valori reale.
(VARIANTA A 2-A) CU UTILIZAREA OPERATORULUI CUT “!”.**

OBSERVAȚIE: Operatorul CUT “!” are rolul de a exclude anumite ramuri de căutare în spațiul soluțiilor în cadrul procesului de backtracking recursiv.

Program PROLOG

```
predicates
    maxim(real, real, real)
    minim(real, real, real)
clauses
    maxim(A, B, A):- A>=B, !.
    maxim(A, B, B).
    minim(A, B, B):- A>=B, !.
    minim(A, B, A).
```

GOAL: maxim(40, 84, Max) minim(-21, -4, Min)

APLICAȚIA 5: Să se determine valoarea funcției:

$f(x,y) = x + y - 2$, dacă $x > -1, y < 1$ și

$f(x,y) = x - y$, în rest.