

o for:

for ( Inic, cond, inc <sup>dec ↓</sup> )  
.....



Em Prolog  
criaram o

(2)

foreach (Var, Lista)

foreach é mais  
simplificado  
mas funciona!

→ Na implementação  
deste predicado ...  
claro que tem recursão  
mas é otimizado!

Funções:

y = função (x<sub>1</sub>, x<sub>2</sub>) ⇒

Em Prolog

função (x<sub>1</sub>, x<sub>2</sub>, y) :-

função sem  
retorno ... ≈ método (Altera o objeto).

procedure (x<sub>1</sub>, x<sub>2</sub>, x<sub>3</sub>, x<sub>4</sub>) = ...

Onde x<sub>3</sub> e x<sub>4</sub> são retorno ⇒

Em Prolog

procedure (x<sub>1</sub>, x<sub>2</sub>, x<sub>3</sub>,  
x<sub>4</sub>) :-

função com função no  
argumento:

Ex: y = f(a, b, g(a, b)) ⇒

Prolog usa  
aceita função como  
argumento!  
Em Prolog.

f(A, B, Y) :-

via  
predicado

.....  
g(A, B, R)  
.....  
Y.....

FUNÇÃO