
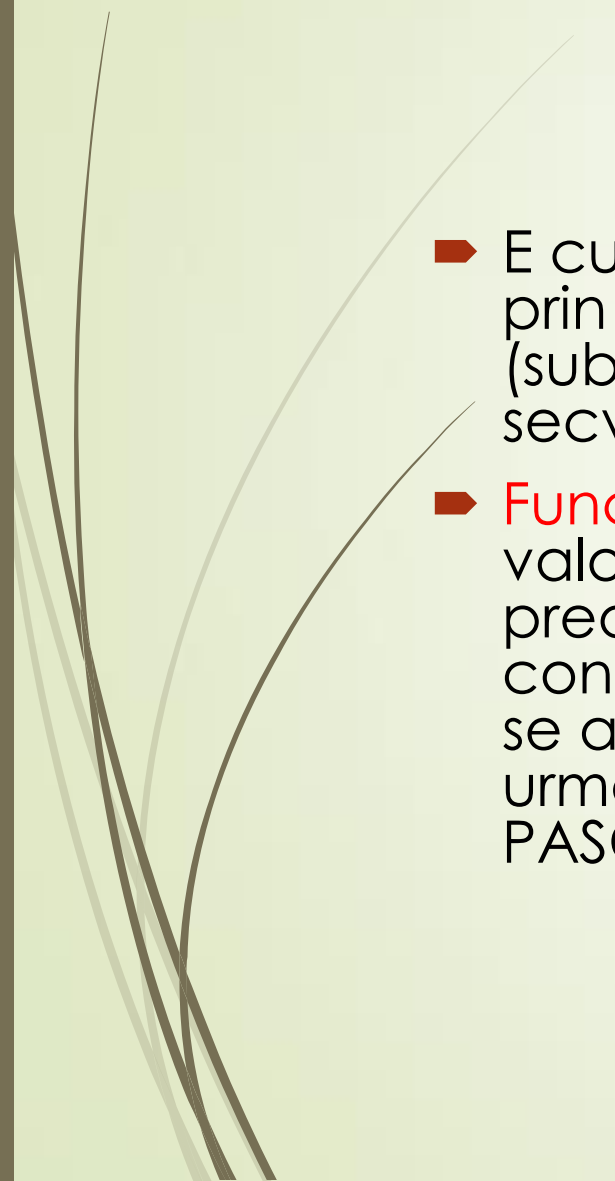


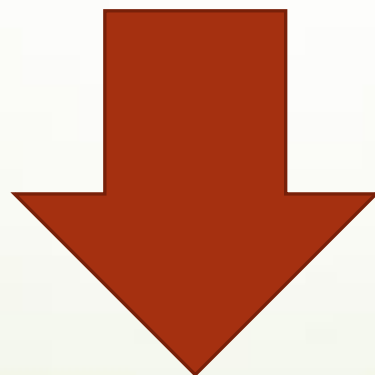
Funcții



- 
- 
- E cunoscut faptul că o problemă complexă poate fi rezolvată prin divizarea ei într-un set de părți mai mici (subprobleme). Pentru fiecare parte se scrie o anumită secvență de instrucțiuni, denumită **subprogram**.
 - **Funcțiile** sînt subprograme care calculează și returnează o valoare. Limbajul PASCAL conține un set de funcții predefinite, cunoscute oricărui program: sin, cos, eof etc. În contemplare, programatorul poate defini funcții proprii, care se apelează în același mod ca și funcțiile-standard. Prin urmare, conceptul de funcție extinde notiunea de **expresie** PASCAL.

Textul PASCAL al unei **declarații de funcție** are forma:

```
Function  $f(x_1, x_2, \dots, x_n)$  :  $t_r$ ;  
D;  
begin  
    ...  
     $f := e$ ;  
    ...  
end;
```





Prima linie este antetul funcției, format din:

- f-numele funcției;
- (x_1, x_2, \dots, x_n) -lista opțională de parametri formali reprezentând argumentele funcției;
- t_r – tipul rezultatului; acesta trebuie să fie numele unui tip simplu sau tip referință.


Antetul este format de **corpul funcției**, format din declarațiile locale opționale D și instrucțiunea compusă **begin...end**.

➤ Declarațiile locale sînt grupate în secțiunile **label, const, type, var, function**.

Numle f al funcției apare cel puțin o dată în partea stîngă a unei instrucțiuni de atribuire care se execută $f:=e$. Ultima valoare atribuită lui f va fi întoarsă în programul principal.



2 exemple
de probleme



1. Să se scrie un program care calculează aria
triunghiului dreptunghic, dacă știm toate laturile.

```
Program Triunghi;  
var a,b,s: real;  
function Aria (a1, b2: real):real;  
var s1 :=(a1*b1)/2;  
  Aria:=S1;  
end;  
begin  
  Write('a='); readln(a);  
  Write ('b='); readln(b);  
  S:=Aria (a,b);  
  Writeln('Aria este',S);  
End.
```

2. Tabel unidimensional: Elementul minim. Funcții

Program Calcul _ min;

Type vector : array [1..100] of integer;

var a:vector;

i,n,min : integer;

function minim (a1:vector; n1:integer):integer;

begin

min1 := a1[1];

for j:1 to n do

if a1[j]<min1 then min1:=a1[j];

minim:=min1;

end;

begin

var min1 j: integer; write('n=');



Cebotari Mihaela clasa X | -a "A"