

Proceduri

Realizat: Mihalachi Mihaela

- **Proceduri** → subprograme care efectuează prelucrarea datelor comunicate în momentul apelului.
- Limbajul conține procedurile predefinite read, readln, write, writeln ș.a.
- În completare, programatorul poate defini proceduri proprii, care se apelează în același mod ca procedurile-standart. Prin urmare, conceptul de procedură extinde noțiunea de instrucțiune PASCAL.

- ◉ **Forma generală** a textului unei declarații de procedură este:

```
Procedure p(x1, x2, ..., xn ) ;  
D;  
begin  
...  
end;
```

- ◉ **p**- numele procedurii;
(x1, x2, ..., xn) –lista opțională de parametri formali;
În corpul procedurii sînt incluse:
D- declarațiile locale(opționale) grupate după aceleași reguli ca în cazul funcțiilor;
Begin...end- instrucțiune compusă; ea nu conține vreo atribuire asupra numelui procedurii.

- Procedura poate să întoarcă mai multe rezultate, dar nu prin numele ei, ci prin variabile desemnate special (cu prefixul var) în lista de parametri formali.
- Parametri-valoare **v1, v2, ..., vk:tp** servesc pentru transmiterea de valori din programul principal în procedură);
- Parametri-variabilă **var v1, v2, ..., vk:tp** (serve pentru întoarcerea rezultatelor din procedură în programul principal);

Exemplu

- De alcătuit un subprogram ce determina suma elementelor pozitive și produsul elementelor negative din $a[1..10]$ of real de folosit procedura.

```
Program P3;
type tab=array[1..10] of real;
var a:tab;
i,n:integer;
s,p:real;
procedure suma(var x:tab;
n:integer; var sum:real);
var i:integer;
begin
s:=0;
for i:=1 to n do
if x[i]>0 then sum:=sum+x[i];
end;
procedure prod(var x:tab;
n:integer; var prod:real);
var i:integer;
begin p:=1;
```

```
for i:=1 to n do if x[i]<0 then
prod:=prod*x[i];
end;
begin{main}
write('n=');
readln(n);
write('dati elementele
tabloului:');
for i:=1 to n do
readln(a[i]);
suma(a,n,s);
write('s=',s);
prod(a,n,p);
write('p=',p);
end.
```