

Liceul Teoretic “Spiru Haret”

Referat la Informatica

“Functii si Proceduri”

Realizat de eleva clasei XI-”B”,

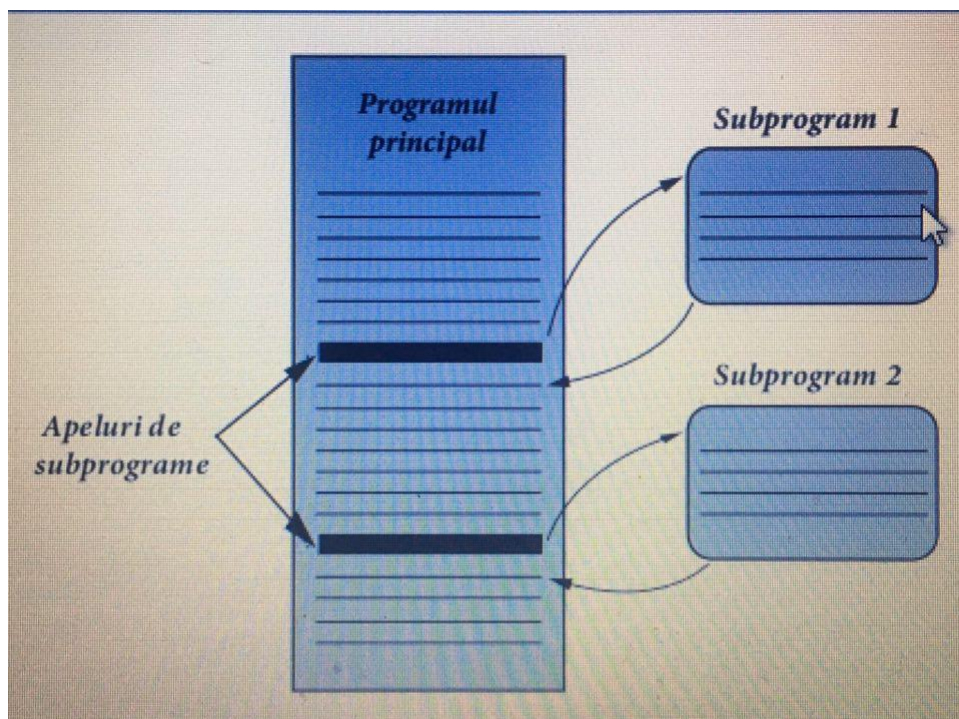
Morari Corina.

Profesor:Gutu Maria

Subprogram

Un *subprogram* este o secventa de instructiuni care poate fi apelata din programul principal sau dintr-un alt subprogram.

Apelul unui subprogram inseamna de fapt lansarea in executie a acelei secvente, ceea ce echivaleaza cu un salt la prima instructiune a subprogramului



În limbajul PASCAL există două tipuri de subprograme:
funcții și proceduri.

Domenii de vizibilitate

Corpul unui program sau subprogram se numește **bloc**. Subprogramele sînt incluse în programul principal și pot conține la rîndul lor alte subprograme, rezulta că blocurile pot fi **imbricate** (incluse unul în altul). Această imbricare de blocuri este denumită **structura de bloc a programului PASCAL**.

Într-o structură fiecărui bloc i se atașează cîte un nivel de imbricare. Programul principal este considerat de nivel 0, un bloc definit în programul principal este de nivel 1. În general, un bloc definit în nivelul n este de nivelul n + 1.

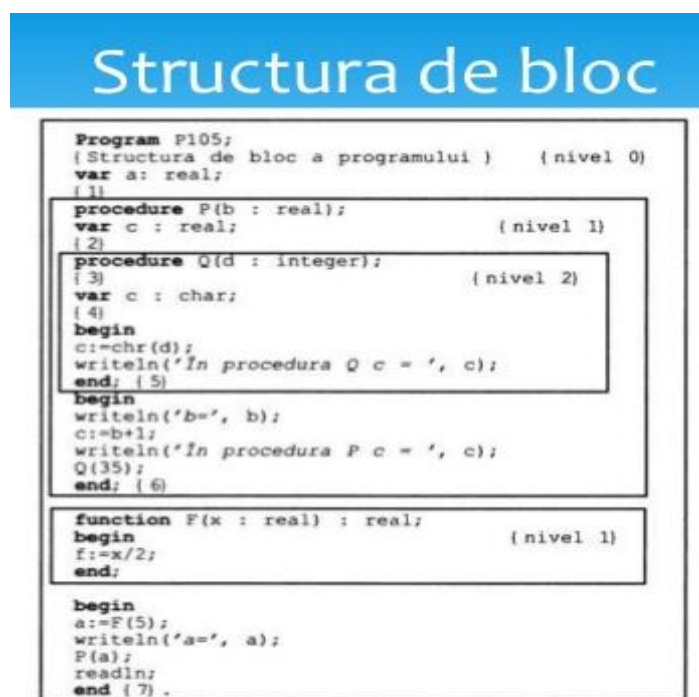


Fig. 5.2. Structura de bloc a unui program PASCAL

O declarație dintr-un bloc poate redefini un nume declarat în exteriorul lui. Deci, în diferite părți ale programului unul și același nume poate desemna obiecte diferite. Prin **domeniul de vizibilitate** al unei declarații se înțelege textul de program, în care numele introdus desemnează obiectul specificat de declarația în studiu. Domeniul de vizibilitate începe imediat după terminarea declarației și se sfârșește odată cu textul blocului respectiv.

Functii

Funcțiile sînt subprograme care calculează și returnează o valoare. Conceptul de funcție extinde noțiunea de expresie PASCAL.

Sintaxa funcției este următoarea:

```
FUNCTION nume ([VAR] l1:t1;[var] l2:t2;...;  
                [var] ln:tn):tip nume;  
[secțiunea declarații locale]  
BEGIN  
{NUME := .... ;} secțiunea instrucțiuni  
END:
```

Exemplul nr1:

Sa se determine distanta dintre doua coordonate.

Program P1;

var a,b,x,y,D:real;

function distanta(a1,b1,x1,y1:real):real;-**Antetul funcției**

Parametri formali valoare

Begin

distanta:=sqrt(sqr(a1-x1)+sqr(b1-y1)); -**corpul funcției/parte executabilă**

end;

begin

readln(a,b,x,y);

D:=distanta(a,b,x,y);

Parametri actuali

```
writeln('distanța dintre coordonate= ',D);
```

End.

Prima linie este antetul funcției, format din:

distanța- **numele funcției**;

(a1,b1,x1,y1) — **lista opțională de parametri formali**
reprezentând argumentele funcției;

Real- **tipul rezultatului**; acesta trebuie să fie e numele unui tip
simplu sau tip referință.

Parametri formali apar în antetul subprogramului și sunt utilizați de subprogram pentru descrierea abstractă a unui proces de calcul .

Parametri actuali apar în instrucțiunea de apelare a unui subprogram și sunt folosiți la execuția unui proces de calcul pentru valori concrete.

Parametrii formali nu sunt variabile. **O variabilă** este caracterizată de nume, tip, și adresă. Legarea unui parametru formal la o adresă se realizează în timpul execuției instrucțiunii de apelare a subprogramului.

Parametrul actual trebuie să fie compatibil din punctul de vedere al atribuirii cu tipul parametrului formal.

Exemplul 2.

Program P2;

{Declararea și utilizarea funcției Putere }

type Natural=0..MaxInt

var a : real;

b : Natural;

c : real;

s : integer;

```

t : integer;
v : real;
function Putere(x : real; n : Natural) : real;
    numele functiei
    { calcularea lui x la puterea n }
var p : real;
    i : integer; parametri formali valoare
begin
p:=1;
for i:=1 to n do p:=p*x; Corpul functiei
Putere:=p;
end;
{ Putere }
begin
a:=3.0;
b:=2;
c:=Putere(a, b);
writeln(a:10:5, b:4, c:10:5);
s:=2;
t:=4;
v:=Putere(s, t); parametri actuali
writeln(s:5, t:4, v:10:5);
readln;
End.

```

Exemplul 3

Type vector=array[1..1000] of integer;

Declararea tabloului unidimensional

```
var a:vector;
```

Atribuirea variabilei unui vector unidimensional

```
var i,n,sum_par:integer;
```

Declararea Variabilelor

```
Function suma:integer;
```

Declararea functiei

```
begin
```

```
sum_par:=0;
```

```
for i :=1 to n do
```

```
if a[i] mod 2=0 then sum_par:=sum_par+a[i];
```

Calcularea sumei numerelor pare

```
suma:=sum_par;
```

Corpul functiei

Atribuirea numelui functiei rezultatului

```
end;
```

```
begin
```

```
Writeln('Introdu nr de termeni');readln(n);
```

```
for i := 1 to n do begin
```

Introducerea datelor in tabloul unidimensional

```
Writeln('A['i,']='); readln(a[i]);
```

```
end;
```

```
suma;
```

```
Writeln('Suma elementelor pare este: ', sum_par);
```

Scoaterea la ecran a sumei

```
readln;
```

```
end.
```

Exemplul 4

Program P4;

{ Redefinirea constantelor }

const c=1;

function F1(x : integer) : integer;

Numele functiei

Begin

F1:=x+c;

end; { F1 }

function F2(c : real) : real;

const x=2.0;

begin

F2:=x+c;

end; { F2 }

function F3(x : char) : char;

const c=3;

Begin

F3:=chr(ord(x)+c);

end; { F3 }

begin

writeln('F1=', F1(1));

writeln('F2=', F2(1));

writeln('F3=', F3('1'));

readln;

end.

Exemplul 5

Program P5;

{ Redefinirea constantelor }

const c=1;

function F1(x : integer) : integer;

Begin

F1:=x+c;

end; { F1 }

function F2(c : real) : real;

const x=2.0;

begin F2:=x+c;

end; { F2 }

function F3(x : char) : char;

const c=3;

begin F3:=chr(ord(x)+c);

end; { F3 }

begin writeln('F1=', F1(1));

writeln('F2=', F2(1));

writeln('F3=', F3('1'));

readln;

end.

writeln(F(1)); { se afișează 1 }

writeln(F(1)); { se afișează 2 }

```
writeln(F(1)); { se afișează 3 }  
readln;  
End.
```

Proceduri

Procedurile sînt subprograme care efectuează prelucrarea datelor comunicate în momentul apelului. Conceptul de procedură extinde noțiunea de instrucțiune PASCAL.

Forma generală a textului unei declarații de procedură este:

```
procedure p(x1, x2, ..., xn);
```

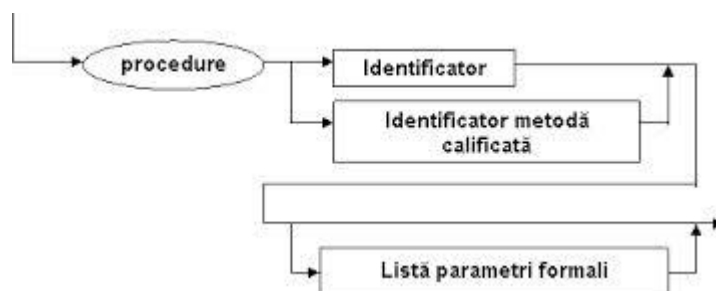
```
D;
```

```
Begin
```

```
...
```

```
end;
```

Diagrama de sintaxa:



Exemplul 6:

Program P6;

Var S,Sa,Stotal,P,ab,bc,cd,ad,ac:real;

Procedure ArieTriunghi(var S1:real; a,b,c:real);

Begin

P:=(a+b+c)/2;

S1:=sqrt(P*(P-a)*(P-b)*(P-c));- instrucțiune compusă;

End;

Writeln('Introdu Datele');

Readln(ab,bc,cd,ad,ac);

ArieTriunghi(S,ab,bc,ac);

ArieTriunghi(Sa,ad,cd,ac);

Stotal:=Sa+S;

Writeln('Aria Patrulaterului= ',Stotal);

End.

În antetul procedurii apar:

ArieTriunghi— **numele procedurii**;

S1,a,b,c— **lista opțională de parametri formali**;

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.

Parametrii introduși prin declarații de forma

$v_1, v_2, \dots, v_k : tp$

se numesc **parametri-valoare**.

Aceștia servesc pentru transmiterea de valori din programul principal în procedură.

Parametrii formali introduși în listă prin declarații de forma

$var\ v_1, v_2, \dots, v_k : tp$

se numesc **parametri-variabilă**

și servesc pentru întoarcerea rezultatelor din procedură în programul principal.

Activarea unei proceduri se face printr-un apel de forma $p(a_1, a_2, \dots, a_n)$

unde a_1, a_2, \dots, a_n

este lista de **parametri actuali**.

În cazul unui **parametru-valoare** drept parametru actual poate fi utilizată orice expresie de tipul respectiv, în particular o constantă sau o variabilă. Modificările parametrilor-valoare nu se transmit în exteriorul subprogramului.

În cazul unui **parametru-variabilă** drept parametri actuali pot fi utilizate numai variabile. Evident,

modificările parametrilor în studiu vor fi transmise programului apelant.

Exemplul 7

```
program DivizoriComuni;
```

```
type Natural=0..MaxInt;
```

```
var a,b,c: Natural;
```

```
procedure DivizComuni(a,b,c:Natural);
```

```
    Numele procedurii
```

```
var i: Natural;
```

```
begin
```

```
  i:=1;
```

```
  while (i*i<=a) do
```

```
    begin
```

```
      if (a mod i =0) then -- instrucțiune compusă;
```

```
        begin
```

```
          if (b mod i=0) and(c mod i=0) then write(i, ' ');
```

```
          if (b mod (a div i)=0)and(c mod(a div i)=0)
```

```
          then write(a div i, ' ');
```

```
        end;
```

```
      i:=i+1;
```

```
    end;
```

```
writeln;  
end;  
begin  
write(' introdu trei numere naturale: ');  
read(a,b,c);  
writeln('Divizorii lor comuni sunt: ');  
DivizComuni(a,b,c);  
End.
```

Exemplul 8

```
Program P3;  
{Declararea și utilizarea procedurii Lac }  
var a, b, c, t, q : real;  
procedure Lac(r : real; var l, s : real);
```

Numele procedurii

```
{lungimea și aria cercului }  
{r - raza; l - lungimea; s - aria }  
const Pi=3.14159;
```

Begin

```
l:=2*Pi*r;
```

```
s:=Pi*sqr(r);
```

- instrucțiune compusă

```
end; {Lac }
```

```
Begin
  a:=1.0;
  Lac(a, b, c);
  writeln(a:10:5, b:10:5, c:10:5);
  Lac(3.0, t, q);
  writeln(3.0:10:5, t:10:5, q:10:5);
  readln;
  End.
```

Exemplul 4

```
program patruNumere;
type Natural = 0..MaxInt;
var a,b,c,d, t, q, s: Natural;
procedure ordonare(var a,b,c,d:Natural);
```

Numele functiei

```
var creste:boolean;
  t: Natural;
begin
  creste:=false;
  while (not creste) do
  begin
    creste:=true;
```

if $a > b$ then begin creste:=false;

t:=a;

a:=b;

b:=t;

end;

if $b > c$ then begin creste:=false;

t:=b;

b:=c;

c:=t;

end;

if $c > d$ then begin creste:=false;

t:=c;

c:=d;

d:=t;

end;

end;

end;

begin

writeln(' introdu 4 numere naturale: ');


```
read(a,b,c,d);  
ordonare(a,b,c,d);  
writeln(a,' ',b,' ',c,' ',d);  
end.
```

Comunicarea prin variabile globale

Orice variabilă este **locală** în subprogramul în care a fost declarată.

O variabilă este **globală** relativ la un subprogram atunci când ea se declară în programul sau subprogramul ce îl cuprinde fără să fie redeclarată în subprogramul, cât și în afara lui, ele pot fi folosite pentru transmiterea datelor de prelucrat și returnarea rezultatelor.

Cuprins:

- Subprogram
- Domenii de vizibilitate
 - Functii
 - Proceduri
- Comunicarea prin variabile globale

Referinte:Cartea de Informatica clasa 11;

https://www.slideshare.net/m_gutu/functii-in-pascal?fbclid=IwAR15a6v-41BkPpIYxGTcQsU3rgT3NFZUWD9vzshZJ-s6NV5UmX7nK9z_TJU;

[https://ru.scribd.com/doc/208824310/Proceduri-Si-Functii-Proceduri-](https://ru.scribd.com/doc/208824310/Proceduri-Si-Functii-Proceduri-Pascal)

[Pascal;https://turbopascal.weebly.com/func539ii.html](https://turbopascal.weebly.com/func539ii.html);

https://turbopascal.ucoz.com/index/proceduri_si_functii/0-8;<https://brainly.ro/tema/5566251>;

<https://brainly.ro/tema/6484917>.