Referat  
 la informatică

Tema: Funcții și proceduri

Elaborat de: Jantoan Nicoleta  
 clasa a XI-a „B”  
 Profesor: Guțu Maria

**CUPRINS**

1. **Funcții- informație generală............................................................3**
2. **Exemple de funcții............................................................................3-6**
3. **Proceduri- informație generală........................................................7**
4. **Exemple de proceduri.......................................................................7-10**
5. **Domenii de vizibilitate......................................................................10**

**FUNCȚII**

**Funcțiile** sunt subprograme care calculează și returnează o singură valoare. Limbajul PASCAL conține un set de funcții predefinite, cunoscute oricărui program: sin, cos, eof etc. În completare, programatorul poate defini funcții proprii, care se apelează în același mod ca și funcțiile-standard.

La realizarea unui program ce conține un subprogram „funcție” ținem cont de :

* Numărul de caractere să fie același ;
* Tipul parametrilor să coincidă ;
* Tipul funcției trebuie să coincidă cu tipul variabilei.

Function <Nume> (x1, x2… xn: <tip1>) : <tip2> Antetul funcției

Parametri formali

<partea declarativa> se declară variabilele locale Corpul

begin funcției

<corpul funcției> secvența de instrucțiuni ce duce

end; la realizarea sarcinii  
 construcția: <Nume> := <expresie>

**EXEMPLE:**

Program N1; nivel 0  
type Natural = 0.. MaxInt;  
var a,c,v: real;  
 b: Natural; variabile globale  
 s, t: integer;

Function Putere (x:real ; n:Natural) : real; nivel 1  
parametri formali antetul functiei  
var p: real; variabile locale  
 i: integer;   
begin

p:=1;  
for i:=1 to n do p:= p\*x;  
Putere:= p; rezultat prin numele ei  
end;  
begin  
a:=3.0;  
b:=2;  
c:= Putere (a,b); parametri actuali  
writeln (a:10:5, b:4, c:10:5);  
s:= 2;  
t:= 4;  
v:= Putere (s,t);  
writeln (s:5, t:4, v:10:5);  
readln;  
end.

Notă!  
Parametrii formali (x, n) se împart în două categorii:

* Parametri formali valoare;
* Parametri formali variabilă (au „var” în față)

Program N2; nivel 0  
var x, y: integer ;  
function F (a : integer) : integer ; indică tipul rezultatului nivel1

numele functiei parametru formal tipul de date

begin  
F := a mod 10; corpul funcției  
a:= a+1;  
end;  
begin  
x := 11; y:=12;  
writeln (‘rezultat:’, F(x)+F(x)+F(y));  
end.

Program N3; nivel 0  
var a,b,c,s: integer;  
function Suma (a1,b1,c1 : integer) : integer; tipul rezultatului nivel1

Numele f-tiei parametri tipul de date  
 formali  
begin  
Suma := a1+b1+c1 ; corpul funcției  
end ;  
begin  
readln (a,b,c);  
S:= Suma (a,b,c); parametrii actuali  
writeln (‘S=’, s);  
end.

Program N4; nivel 0  
var l,a,r,pi: real;  
function ariecerc (pi1, r1 : real) : real; tipul rezultatului nivel 1

Numele f-tiei parametri tipul de date  
 formali  
begin  
ariecerc := pi1\*sqr(r1) ; corpul funcției  
end ;  
function lungimecerc (pi1 ; r1 :real) :real ; tipul rezultatului nivel 1

Numele f-tiei parametri formali tipul de date  
begin  
lungimecerc := 2\*pi1\*r1 ;  
end ;  
begin  
readln (r) ;  
pi := 3,14;  
l:= lungimecerc (pi,r);  
a:= ariecerc (pi, r); parametrii actuali  
writeln (‘l=’, l);  
writeln (‘a=’,a);  
end.

Program N5;  
type Vector = array [1..100] of real;  
var I, numbers : integer;  
var numbersA: Vector;  
function produs (a: Vector; numbers: integer): real; tipul rezultatului

Numele functiei parametrii tipurile de date  
 formali  
var total : real ; partea declarativă  
var i : integer ;  
begin  
total :=1 ;  
for i := 1 to numbers do corpul funcției  
if a [i]<0 then total:= total\*a[i];  
produs := total;  
end;  
begin  
writeln (‘Coeficientul de numere:’);  
readln (numbers);  
for i:= 1 to numbers do  
begin  
writeln (‘introduceti un numar:’);  
readln (numbersA[i]);  
end;  
writeln (produs (numbersA, numbers):2:2);  
end.

**Proceduri**

**Procedurile** sunt subprograme care efectuează prelucrarea datelor communicate în momentul apelului. Procedura poate întoarce mai multe rezultate, prin variabile desemnate (cu prefixul var) în lista parametrilor formali.

* var v1, v2... vn : <tip1> - parametri-variabilă  
   (întoarcerea rezultatelor din procedură în programul principal)
* v1, v2… vn: <tip1>- parametric-valoare  
  (transmiterea de valori din programul principal în procedură)

Notă!

* Parametru-valoare – drept parametru actual pot fi luate expresii, constant sau variabile.
* Parametru-variabilă – parametrii actuali sunt numai variabile.

Procedure <Nume> (x1,x2… xn); antetul

Lista opțională de parametri formali procedurii

<partea\_declarativa>

begin Corpul

<corpul\_procedurii> procedurii

end;

**EXEMPLE**

Program N6; nivel 0  
var a,b,c,t,q: real; variabile globale  
procedure Lac (r:real; var l,s :real); nivel 1

Parametru formal valoare parametri formali-variabila  
const Pi = 3,14159;  
begin  
l:= 2\*Pi\*r;  
s:= Pi\*sqr (r);  
end;  
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.

Program N7; nivel 0  
var x,y,a,p :real; variabile globale  
procedure Ariedr (x1, y1: real; var a1:real); antet nivel1

Parametrii formali-valoare parametru formal variabila

begin  
a1:= x1\*y1; corpul procedurii  
end;  
procedure Perimetrudr (x1,y1: real; var p1:real); antet nivel 1

parametrii formali valoare parametrul formal variabila  
begin  
p1 := 2\*(x1+y1); corpul procedurii  
end;  
begin  
write (‘x=’); readln (x);  
write (‘y=’); readln (y);  
Ariedr (x,y,a);  
Perimetrudr (x,y,p);  
writeln (‘a=’, a);  
writeln (‘p=’,p);  
end.

Program N8; nivel 0  
Type Vector= array [1..100] of integer;  
var a: vector;  
 p,i,n :integer;  
procedure citire (n1: integer; var a1: vector); antet nivel 1  
var i: integer; parametru-valoare parametru-variabila  
begin elementul tabloului  
for i:= 1 to n1 do readln (a1[i]); pozitia

end;  
procedure Produs (b1:Vector; var p1: integer); nivel 1  
var i: integer; partea declarativă  
begin  
p1:=1;  
for i:=1 to n do corpul procedurii  
if (b1[i] mod 2=0) and (i mod 2=1) then p:=p\*b1[i];  
end;  
begin  
readln (n);  
Citire (n, a);  
Produs (a, p);  
writeln (‘p=’,p);  
end.

Program N9; nivel 0  
var s, stotal, ab, bc, cd, ad, ac :real; partea declarativa  
procedure Arietriunghi (var s1 : real; antet nivel 1  
 var b,c :real);  
begin  
p:= (a+b+c)/2; corpul procedurii  
s1:=sqrt(p\*(p-a)(p-b)(p-c));  
end;  
begin  
writeln (‘Introduceti datele’);  
readln (ab, bc, cd, ad, ac);  
Arietriunghi (s, ab, bc, ac);  
Arietriunghi (sa, ad, cd, ac);  
stotal:= sa+s;  
writeln (‘stotal=’, stotal);  
end.

Program N10; Nivel 0  
var a1, b1, c1, m :real;  
procedure maxim (a,b,c: real; var f:real); antet nivel 1  
var max: real; partea declarativa  
begin  
max := a;  
if max<b then max:=b else max:=a;  
if max<c then max:=c;  
f:=max;  
end;  
begin  
writeln (‘Dati variabilele’);  
readln (a1,b1,c1);  
maxim (a1,b1,c1,m);  
writeln (‘Maximul este’,m);  
readln;  
end.

**Domenii de vizibilitate**

APELUL SUBPROGRAMELOR

1. Programul principal nu poate fi apelat de nici un bloc.
2. Orice bloc poate să apeleze blocurile proprii.
3. Nici un bloc nu poate să apeleze subprogramul subprogramului său.
4. Dacă sunt 2 subprograme de același nivel, atunci blocul situat mai jos poate apela blocul situat mai sus.
5. Orice subprogram se poate autoapela (recursie).