

## Funcțiile

Textul Pascal al unei declarații de funcție are forma:

```
function f(x1, x2, ..., xn) : tr;
```

```
D;
```

```
begin
```

```
...
```

```
f := e;
```

```
...
```

```
end;
```

Prima linie este antetul funcției, format din:

f — numele funcției;

(x1, x2, ..., xn) — lista opțională de parametri formali reprezentând argumentele funcției;

tr

— tipul rezultatului; acesta trebuie să fie numele unui tip simplu sau tip referință.

7

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

Declarațiile locale sînt grupate în secțiunile (eventual vide) label, const, type, var, function/procedure.

Numele 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.

În mod obișnuit, un parametru formal din lista (x1, x2, ..., xn) are forma:

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

unde  $v_1, v_2, \dots, v_k$  sînt identifi catori, iar  $tp$  este un nume de tip.

Utilizarea funcției  $f$  se specifi că printr-un apel de forma:

$f(a_1, a_2, \dots, a_n)$

unde  $(a_1, a_2, \dots, a_n)$  este lista de parametri actuali. De obicei, parametrii actuali sînt expresii,

valorile căroră sînt comunicate funcției. Corespondența între un parametru actual și parametrul formal se face prin poziția ocupată de aceștia în cele două liste.

Parametrul actual

trebuie să fi e compatibil din punctul de vedere al atribuirii cu tipul parametrului formal.

Exemplu:

Program P97;

{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;

{ calcularea lui x la puterea n }

var p : real;

i : integer;

begin

p:=1;

```

for i:=1 to n do p:=p*x;
Putere:=p;
end; { Putere }

begin
a:=3.0;
b:=2;
c:=Putere(a, b);
writeln(a:10:5, b:4, c:10:5);
8
s:=2;
t:=4;
v:=Putere(s, t);
writeln(s:5, t:4, v:10:5);
readln;
end.

```

Funcția Putere are doi parametri formali: x de tipul real și n de tipul Natural.

Funcția returnează o valoare de tipul real. În corpul funcției sînt declarate variabilele locale p și i.

La execuția apelului Putere(a,b) valorile 3.0 și 2 ale parametrilor actuali a, b se transmit parametrilor formali, respectiv, x și n. De menționat că tipul lui a coincide cu tipul lui x și tipul lui b coincide cu tipul lui n.

În cazul apelului Putere(s,t) tipul parametrilor actuali s,t nu coincide cu tipul parametrilor formali, respectiv, x și n. Totuși apelul este corect, întrucît tipurile

respectve sînt compatibile din punctul de vedere al atribuirii.

## Proceduri

Forma generală a textului unei declarații de procedură este:

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

```
  D;
```

```
begin
```

```
...
```

```
end;
```

În antetul procedurii apar:

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 și în cazul funcțiilor;

11

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 prefîxul var) în lista de parametri formali.

Parametrii din listă introduși prin declarații de forma

```
v1, v2, ..., vk : 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 v1, v2, ..., vk : 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(a1, a2, ..., an)
```

unde  $a_1, a_2, \dots, a_n$  este lista de parametri actuali. Spre deosebire de funcție, apelul de procedură este o instrucțiune; aceasta se inserează în programul principal în locul în care sînt dorite efectele produse de execuția procedurii.

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.

Exemplu:

Program P99;

{Declararea și utilizarea procedurii Lac }

var a, b, c, t, q : real;

procedure Lac(r : real; var l, s : real);

{lungimea și aria cercului }

{r - raza; l - lungimea; s - aria }

const Pi=3.14159;

begin

l:=2\*Pi\*r;

s:=Pi\*sqr(r);

end; {Lac }

begin

a:=1.0;

Lac(a, b, c);

writeln(a:10:5, b:10:5, c:10:5);

12

```
Lac(3.0, t, q);  
writeln(3.0:10:5, t:10:5, q:10:5);  
readln;  
end.
```

Procedura Lac are trei parametri formali: r, l și s. Parametrul r este un parametru-valoare, iar l și s sînt parametri-variabilă.

Execuția instrucțiunii Lac(a,b,c) determină transmiterea valorii 1.0 drept valoare a parametrului formal r și a locațiilor (adreselor) variabilelor b și c drept locații (adrese) ale parametrilor formali l și s. Prin urmare, secvența de instrucțiuni

```
a:=1.0;
```

```
Lac(a, b, c)
```

este echivalentă cu secvența

```
b:=2*Pi*1.0;
```

```
c:=Pi*sqr(1.0).
```

În mod similar, instrucțiunea

```
Lac(3.0,t,q)
```

este echivalentă cu secvența

```
t:=2*Pi*3.0;
```

```
q:=Pi*sqr(3.0).
```