**Referat la informatică**

**Tema: Funcții și proceduri**

Elaborat de: Monastîrli Sanda

Clasa a XI-a ”B”

Profesor: Guțu Maria

**CUPRINS**

1. Subprograme……………………………………………………....3
2. Funcții………………………..……………………………………3
3. Exemple de funcții………………………………………………4-7
4. Proceduri……………………...……………………………………8
5. Exemple de proceduri………………………………………….8-14

**Subprograme**

O problemă complexă poate fi rezolvată prin divizarea ei într-un set de părți mai mici. Pentru fiecare parte se scrie o anumită secvență de instrucțiuni, denumită subprogram.

**Funcții**

Funcțiile sunt subprograme care calculează și returnează o valoare. Limbajul PASCAL conține un set de funcții predefinite, cunoscute oricărui program : sin, cos, eof etc.

În completare, programul poate defini funcții proprii, care se apelează în același mod ca funcțiile-standart. Prin urmare, conceptul de funcție extinde noțiunea de expresie PASCAL.

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

1. Numărul de caractere să fie același
2. Tipul parametrilor să coincidă
3. Tipul funcției trebuie să coincidă cu tipul variabilei

Function NumeFunctie([Var] pf11[,pf12]..]:tip1;[Var] [pf21[,pf22]..:tip2]

                        ....[;[Var] pfn1[,pfn2]..:tipn]):TipFunctie;

Const {declaratii de constante locale functiei}

Type {declaratii de tipuri locale functiei}

Var {declaratii de variabile locale functiei}

[Function...... {descrierea functiilor apelate de functia curenta};]

Begin {descrierea algoritmului functiei)

End.

**Exemple:**

**Program P1;**

type tab=array[1..10] of real;

var a:tab;

i,n:integer;

s:real;

function suma(x:tab; n:integer):real;

var i:integer;

z:real;

begin

z:=0;

for i:=1 to n do

z:=z+x[i];

suma:=z;

end;

begin{main}

write('n='); readln(n);

write('dati elementele tabloului:');

for i:=1 to n do readln(a[i]);

s:=suma(a,n);

write('s=',s:5:2);

**end.**

**Program P2;**

type tab=array[1..10] of real;

var a:tab;

locul,i,n:integer;

min:real;

function minim(var x:tab; k:integer):real;

var j:integer;

begin

min:=x[1]; locul:=1;

for j:=2 to k do begin

if x[j]<min then

begin

min:=x[j]; locul:=j;

end;

end;

minim:=min;

end;

begin

write('n='); readln(n);

for i:=1 to n do readln(a[i]);

min:=minim(a,n);

writeln('min=',min:5:2,'locul=',locul);

**end.**

**Program P3;**var h1,h2, min1, min2, sec1, sec2, T1, T2, Tdiferenta : integer;

function Timpul (h, min, sec: integer) : integer;  
begin  
Timpul :=h\*3600+min\*60+sec;  
end;  
begin  
writeln (‘Introduceti datele’);  
readln (h1, min1, sec1);  
readln (h2, min2, sec2);  
T1 := Timpul (h1, min1, sec1);  
T2 := Timpul (h2,min2,sec2);  
Tdiferenta := abs (T1-T2);  
writeln (‘Diferenta dintre cele doua momente este’,Tdiferenta);  
**end.**

**Program P4;**var x, y, z, a, d : real;  
function Dist (x1,y1,z1,a1 :real) : real;  
begin  
d := sqrt (sqr (x1-a1) + sqr (y1-z1));  
end;  
begin  
writeln (‘Introduceti coordonatele primului punct’);  
readln (x,y);  
writeln (‘Introduceti coordonatele punctului doi’);  
readln (a,z);  
d := Dist (x,y,z,a);  
writeln (‘Distanta dintre coordonate este’,d);  
**end.**

**Program P5;**

Var a, b, suma, diferenta, produs, cat:real;

raspuns:char;

Function fsuma(x, y:REAL;VAR difxy, produsxy, catxy:real):real;

Begin

difxy:=x-y; produsxy:=x\*y; catxy:=x/y;

fsuma:=x+y;

End;

Begin

raspuns:= 'd';

While (raspuns= 'd') Do

Begin

Write ('dati nr reale a, b:'); Readln (a, b);

If (b<>0) Then

Begin

suma:= fsuma(a, b, diferenta, produs, cat);

Writeln ('a+b=', suma:1:2, ' a-b=', diferenta:1:2);

Writeln ( 'a\*b=', produs:1:2, ' a/b=', cat:1:2);

End

Else

Writeln ('impartire prin b=0, apel imposibil de functie');

Write ('continuati ?(d/n):'); Readln(raspuns);

End;

**End.**

**Proceduri**

Procedurile sînt 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.

Procedure NumeProcedura ([Var ]pf11[,pf12]...:tip1]

[;[Var ]pf21[,pf22]...:tip2]...[;[Var ]pfn1[,pfn2]..]:tipn]);

Const {declaratii de constante locale procedurii}

Type {declaratii de tipuri locale procedurii}

Var {declaratii de variabile locale procedurii}

Begin {descrierea subalgoritmului procedurii}

End.

**Exemple:**

**Program P1;**

type tab=array[1..10] of real;

var a:tab;

i,n:integer;

s:real;

procedure suma(var x:tab; n:integer; var sum:real);

var i:integer;

begin

sum:=0;

for i:=1 to n do

sum:=sum+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:7:2);

**end.**

**Program P2;**

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:5:2);

prod(a,n,p);

write('p=',p:5:2);

**end.**

**Program P3;**

type tab=array[1..10] of integer;

var a:tab;

i,k,n:integer;

procedure citire(var b:tab; var k:integer);

var i:integer;

begin

write('k='); readln(k);

write('tastati','k','numere:');

for i:=1 to k do read(b[i]);

end;

procedure afisare(var b:tab; k:integer);

var i:integer;

begin

writeln('continutul tabloului:');

for i:=1 to k do

write(b[i],' ');

end;

procedure inlocuire(var x:tab; n:integer);

var i:integer;

begin

for i:=1 to n do

if x[i] mod 2=1 then x[i]:=0

else x[i]:=1;

end;

begin{main}

citire(a,n);

inlocuire(a,n);

write('tabloul obtinut:');

afisare(a,n);

**end.**

**Program P4;**  
var S, Stotal, ab, bc, cd, ad, ac :real;  
procedure Arietriunghi (var S1: real; var b,c : real);  
begin  
p := (a+b+c)/2;  
S1 := sqrt (p\*(p-a)(p-b)(p-c));  
end;  
begin  
writeln (‘Introdu 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 P5;**

type Ora = 0..23;

Grade = -40..40;

Temperatura = array [Ora] of Grade;

var t: Temperatura;

n : integer;

m, maxim, minim : real;

procedure TMed (var med : real);

var s : real; i,k : integer;

begin

s := 0;

k := 0;

for i := 0 to n do

begin

s := t[i] + s;

inc (k);

med := s/k;

end;

procedure Max (var maxim:real);

var i : integer;

begin

maxim := t[0];

for i := 1 to n do

if t[i] > maxim then maxim := t[i];

end;

Procedure Min (var minim : real);

begin

minim := t[0];

for i := 1 to n do

if t[i] < minim then minim := t[i];

end;

procedure OraMax (maxim :real);

var i : integer;

begin

for i := 0 to n do

if t[i] = maxim then writeln ('Temperatura maxima s-a inregistrat la ora',i);

end;

procedure OraMin (minim: real);

var i : integer;

begin

for i := 0 to n do

if t[i] = minim then writeln (' Temperatura minima s-a inregistrat la ora',i);

end;

procedure Citire

var i : integer;

begin

for i := 0 to n do read (t[i]);

end;

procedure Afisare;

var i : integer;

begin

for i := 0 to n do

write (t[i]:3);

end;

begin

write ('n=');

n := 5;

readln (n);

Citire;

Afisare;

TempMed (m);

Max (maxim);

Min (minim);

OraMax (maxim);

OraMin (minim);

**end.**