IPLT “Spiru Haret”

Referat

**Funcţii şi proceduri**

**Elev: Alina Maria Ţurcanu**

**Clasa: 11 „B”**

**Chişinău**

**2020**

Cuprins

[Subprograme 3](#_Toc39320095)

[Domeniul de vizibilitate 3](#_Toc39320096)

[Funcții 4](#_Toc39320097)

[Exemple de probleme 5](#_Toc39320098)

[Proceduri 7](#_Toc39320099)

[Exemple de probleme 8](#_Toc39320100)

[Referințe bibliografice: 11](#_Toc39320101)

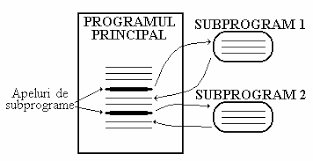
## Subprograme

Înainte de studierea aprofundată a funcţiilor şi procedurilor, trebuie de menţionat că acestea sunt nişte **subprograme.**

Un **subprogram** este o serie de instrucţiuni care pot fi apelate din programul principal sau din alte subprograme.

De asemenea, reprezintă un ansamblu ce poate conţine tipuri de date, variabile şi instrucţiuni destinate unei anumite prelucrări (calcule, citiri, scrieri). Subprogramul poate fi executat doar dacă este apelat de către un program sau un alt subprogram.

**În limbajul Pascal, subprogramele sunt de 2 tipuri: funcţii şi proceduri.**

****

## Domeniul de vizibilitate

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

**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 sfîrșește o dată cu textul blocului respectiv.Domeniul de vizibilitate al unei declarației inclus acoperă  domeniul de vizibilitate al declarației ce implică același nume din blocul exterior.**

**VARIABILE GLOBALE ȘI LOCALE – o variabilă este globală relativ la un subprogram atunci când ea este declarată în program sau subprogramul ce îl cuprinde, fără să fie redeclarată în subprogramul în studiu. Se utilizează în cazul în care mai multe subprograme prelucrează aceleași date.**

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

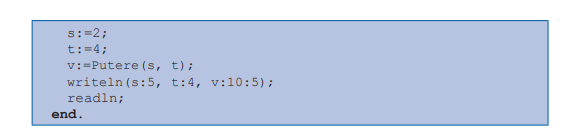
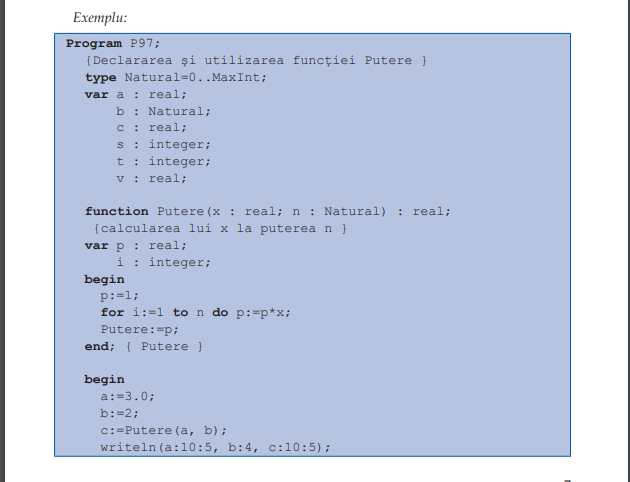
## Funcții

Funcțiile sunt niște subprograme sau niște grupuri auxiliare, care ne ajută să executăm de mai multe ori aceleași comenzi din programul principal, fără să le repetăm. Altfel spus, **funcțiile sunt subprograme care calculează și returnează o valoare.**

**PARAMETRI**

variabila

valoare

Am luat un exemplu din manual, pagina 7-8, pentru a arăta urmatoarele elemente:

Parametri actuali

Variabile locale

Parametri formali, tipul valorii returnate

Variabile globale

# Exemple de probleme

1. **Se considera cuvintul S. Să se scrie un program care inverseaza cuvantul dat.**

program p1;

Declaram o functie PR1 cu un singur argument S de tip STRING care returneaza un cuvant. function pr3(s: string): string;

function pr1(s: string): string;

var i: integer; var c: char;

begin

for i := 1 to length(s) div 2 do

begin

Declaram 2 variabile locale intregi: c – pentru o litera auxiliara, i – pentru a parcurge variabila S.

c := s[i];

s[i] := s[length(s) - i + 1];

s[length(s) - i + 1] := c;

end;

pr1 := s;

Parcurgem cuvantul pana la jumatate. Litera curenta s[i] o vom schimba cu locul cu litera opusa (de la sfarsitul cuvantului)

end;

Functia va returna cuvantul inversat

var p: integer;

var s: string;

begin

writeln; writeln('Dati un cuvant: ');

readln(s);

s:= pr1(s);

writeln(s);

readln;

end.

1. **Se considera sirul S. Să se scrie un program care claculează câte vocale sunt în sir (utilizând subprograme).**

program p2;

Declaram o functie vocale cu un singur argument S de tip STRING care returneaza un numar intreg.

function vocale(s: string): integer;

var v, i: integer;

begin

for i := 1 to length(s) do

Declaram 2 variabile locale intregi: V – pentru numarul de vocale, I – pentru a parcurge variabila S.

if upcase(s[i]) in ['A', 'E', 'I', 'O', 'U'] then v := v + 1;

vocale:=v; end;

var p: integer;

var s: string;

begin

Parcurgem cuvantul si verificam fiecare litera. Fiecare litera o vom transforma in litera mare (cu functia upcase) si o vom verifica daca se include in multimea de vocale ['A', 'E', 'I', 'O', 'U']. Daca se include, incrementam variabila V.

s:='Functie';

writeln;

Functia va returna numarul de vocale gasite

writeln('In cuvantul ', s, ' sunt ', vocale (s), ' vocale.');

readln;

end.

1. **Se considera cuvântul S. Să se scrie un program care claculează ce sunt mai multe vocale sau consoane (utilizând subprograme).**

Program P3;

Var n,m,i:integer;

Function voccons(s:string):integer;

Var i:integer;

Begin

For i:=1 to length (s) do

If s[i] in [`i`,`o`,`u`,`y`,`e`,`a`] then n:=n+1;

For i:1 to length (s) do

If s[i] in [`q`,`w`,`r`,`t`,`p`,`s`,`d`,`f`,`g`,`h`,`j`] then

m:=m+1;

end;

begin

readln(s);

voccons(s);

if m>n then writeln (`mai multe consoane`) else

writeln (`mai multe vocale`);

end.

1. **Se considera tabloul unidimensional a[1..10]of real. Sa se determine suma elementelor acestui tablou printr-un subprogram.**

Program P2;

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

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.

1. **Sa se scrie un program care citeste de la tastatura doua perechi de numere (x1,x2) si (y1,y2) si sa se calculeze media aritmetica a fiecarei perechi.**

Progam P5;

Var m1,m2,x1,x2,y1,y2:real;

Function calcul(x,y:real):real;

Begin

Calcul:=(x+y)/2;

End;

Begin

Write(’x1,x2=’);

Readln(x1,x2);

M1:=calcul(x1,x2);

Write(’y1,y2=’);

Readln(y1,y2);

M1:=calcul(y1,y2);

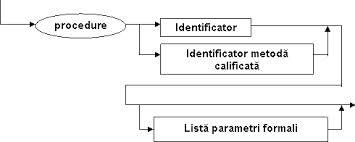
readln;

end.

## Proceduri

**Procedurile sunt subprograme care efectuează prelucrarea datelor comunicate în momentul apelului.**

**Am explicat mai sus elementele caracteristice. Într-o declarație de procedură sunt obligatorii doar antetul de procedură și instrucțiunea compusă BEGIN...END.**



# Exemple de probleme

1. **Se considera cuvântul S. Să se scrie un program care claculează ce sunt mai multe vocale sau consoane (utilizând subprograme).**

Declaram o procedura PR1 cu un singur argument S de tip STRING

program p1;

procedure pr1(s: string);

var v, c, i: integer;

Declaram 3 variabile locale intregi: V – pentru numarul de vocale, C – pentru numarul de consoane, I – pentru a parcurge variabila S

begin

for i := 1 to length(s) do

if upcase(s[i]) in ['A', 'E', 'I', 'O', 'U'] then v := v + 1

else c := c + 1;

if v > c then write('Sunt mai multe vocale')

else if c > v then write('Sunt mai multe consoane')

else write('Numarul de consoane si vocale este egal');

end;

Parcurgem cuvantul si verificam fiecare litera. Fiecare litera o vom transforma in litera mare (cu functia upcase) si o vom verifica daca se include in multimea de vocale ['A', 'E', 'I', 'O', 'U']. Daca se include, incrementam variabila V. Daca nu – incrementam variabila C.

Comparam variabilele C si V si afisam mesajul respectiv.

end.

1. **Se consideră cuvântul S. Să se scriu un program care formează alt cuvânt P cu literele din cuvântul S. (utilizând subprograme)**

Program P2;

Var S,P:string;

Procedure Cuvant(var sir:string);

Var i:integer;

Begin

Sir:=` `;

For i:=1 to length(S) do sir:=sir+S[i];

End;

Begin

Writeln (‘introduceti sirul ‘);

Readln (s);

Cuvant(p);

Writeln (‘sirul obtinut ‘);

Writeln (p);

End.

1. **Se considera cuvantul S. Sa se scrie un program care inverseaza cuvantul dat.**

Program P3;

Procedure inversare (var sir: string);

Var i:integer;

Begin

Sir:=` `;

For i:=l to length (S) do sir:=S[i]+sir;

End;

Begin

Writeln (‘introduceti sirul ‘);

Readln(S);

Inversare (P);

Writeln (‘sirul obtinut: ‘);

Writeln (P);

End.

1. **Se considera cuvantul S. Sa se scrie un program care formeaza alt cuvant P cu literele din cuvantul S fara a utiliza litera `a`.**

Program P4;

Var S,P: string;

Procedure calcul(var sir:string);

Var i:integer;

Begin

Sir:=` `;

For i:=1 to length(S) do if S[i]<>`a` then

Sir:=sir+S[i];

End;

Begin

Writeln (‘Introduceti sirul ‘);

Readln (S);

Calcul (P);

Writeln (`sirul obtinut: `);

Writeln (P);

End.

1. **De la tastatura se introduce un sir de caractere reprezentat de variabila s. Determinati in acest sir de cate ori se intalnesc doua litere consecutive cu aceeasi valoare.**

Program P5;

Var s:string;

K:integer;

Procedure Calcul(var sir:string; var k:integer);

Var i:integer;

Begin

For i:=1 to length(sir)-1 do if sir[i+1]=sir[i] then inc(k);

End;

Begin

Writeln (‘Introduceti sirul’);

Readln(s);

Calcul(s,k);

Writeln(k);

End.

## Referințe bibliografice:

1. Manual de informatică, clasa a 11-a

[file:///C:/Users/Admin/Downloads/XI\_Informatica%20(in%20limba%20romana).pdf](file:///C:\Users\Admin\Downloads\XI_Informatica%20(in%20limba%20romana).pdf)

1. <https://cuochiipe.files.wordpress.com/2016/05/functii-si-proceduri-pdf1.pdf>
2. <https://infopascalfacts.wordpress.com/2016/11/09/proceduri/>
3. <https://sites.google.com/site/stanciu4info4tic/home/clasa-x/parametrii-actuali-si-formali>
4. <https://proiectinformaticamd.wordpress.com/2015/04/21/proceduri-pascal/>