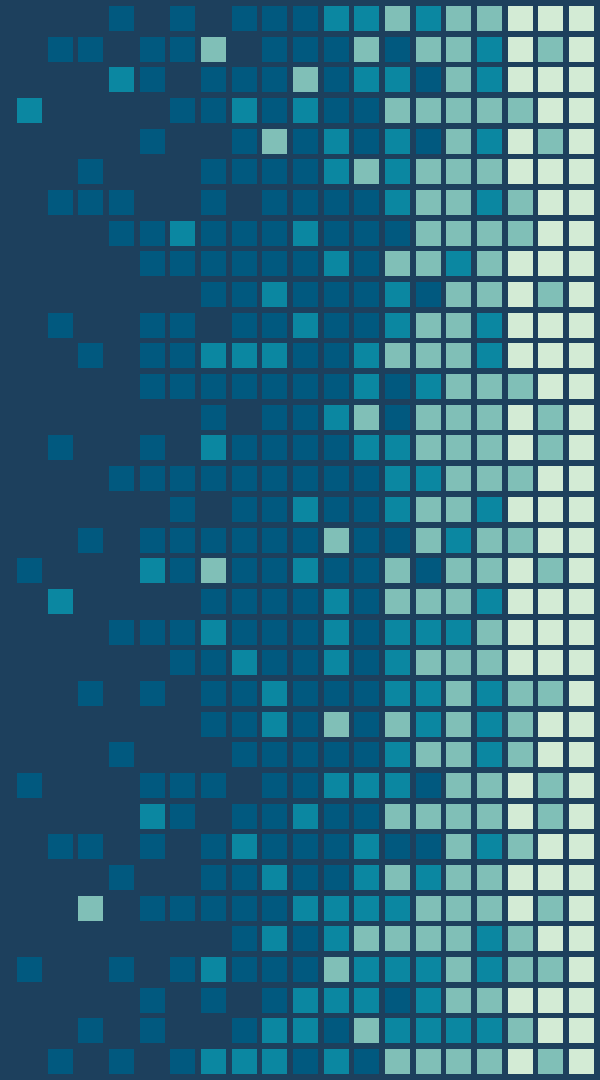
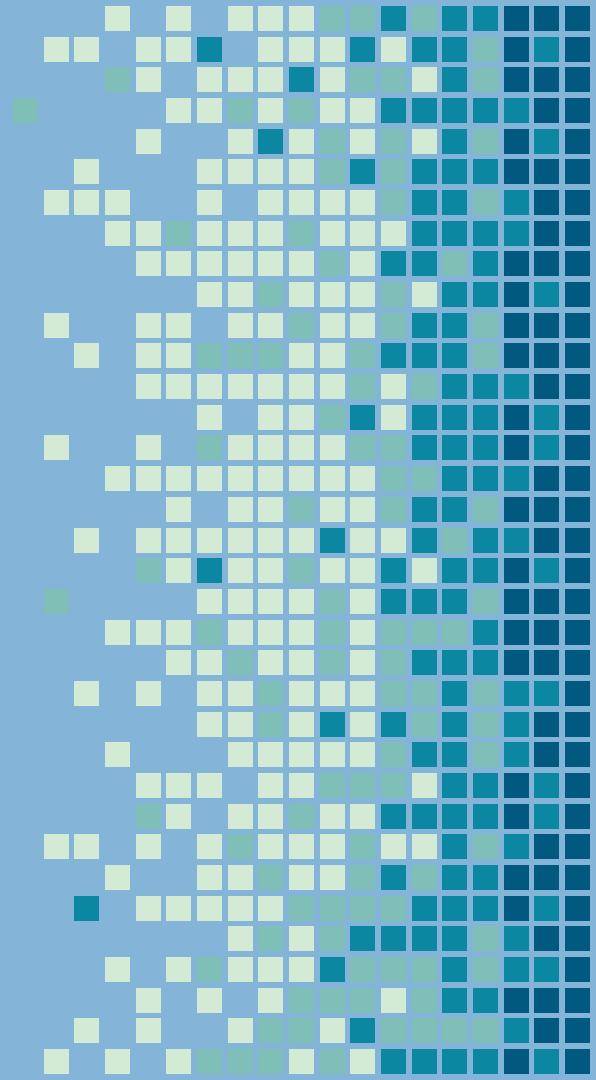


Funcții



Scopuri și obiective

1. Descifrarea noțiunii de *funcție*
2. Observarea formei acesteia
3. Studiarea părților componente
4. Analiza unei funcții definite
5. Cercetarea exemplului



Funcțiile – subprograme ce calculează
și returnează o valoare.

- Limbajul Pascal conține un set de funcții predefinite, cunoscute oricărui program (sin, cos, eof etc).
- Programatorul poate defini funcții proprii, care se apelează în același mod ca și funcțiile-standard. Astfel, conceptul de funcție depășește noțiunea de expresie Pascal.

Sintaxa funcției

```
function name(argument(s): type1; argument(s): type2; ...):  
function type;  
local declarations;  
begin  
...  
< statements >  
...  
name:= expression;  
end;
```



*Textul PASCAL al unei declarații de funcție constă din
antetul funcției și **corpul funcției**.*

Antetul funcției constă din *numele funcției*, *lista opțională de parametri formali* și *tipul rezultatului*.

***Argumentele** stabilesc legătura dintre programul de așteptare și identificatorii de funcții, de asemenea, numiți parametri formali.

Corpul funcției conține o colecție de instrucțiuni care definesc ce face funcția. Ar trebui să fie întotdeauna închisă între începutul și sfârșitul cuvintelor rezervate (begin ... end).

Trebuie să existe o instrucțiune de atribuire a tipului - nume: = expresie; care atribuie o valoare numei funcției.

Declarațiile locale se referă la declarațiile pentru etichete, constante, variabile, funcții și proceduri, care se aplică numai corpului de funcție (label, const, type, var, function/
5procedure).



Exemplu definire a funcției :

```
(* function returning the max between two numbers *) function
max(num1, num2: integer): integer;
var (* local variable declaration *)
result: integer;
Begin
  if (num1 > num2) then result := num1
  else result := num2;
  max := result;
end;
```

Explicații :

Declarații de funcții

O declarație de funcții indică compilatorului despre un nume de funcție și despre modul de apelare a funcției. Organismul real al funcției poate fi definit separat.

O declarație a funcției are următoarele părți :
`function name(argument(s): type1; argument(s): type2; ...): function_type;`

Pentru funcția de mai sus definită `max ()`, declarația funcției este următoarea :
`funcția max (num1, num2: integer): integer;`



Apelarea unei funcții

Pentru a utiliza o funcție, va trebui să apelați această funcție ca să efectueze sarcina definită. Atunci când un program solicită o funcție, programul de control este transferat la funcția apelată. Funcția numită îndeplinește o sarcină definită și atunci când declarația de returnare este executată sau când este atinsă declarația de sfârșit, ea returnează controlul programului înapoi la programul principal.

Pentru a apela o funcție, pur și simplu trebuie să treci parametrii necesari împreună cu numele funcției și dacă ea returnează o valoare, atunci poți stoca valoarea returnată. *În continuare este un exemplu simplu pentru a arăta utilizarea -*



```
program exFunction;
a, b, ret : integer;
(*function definition *)
function max(num1, num2: integer): integer;
var
(* local variable declaration *)
result: integer;
begin
if (num1 > num2) then result := num1 else result := num2; max := result;
end
begin
a := 100;
b := 200;
(* calling a function to get max value *)
ret := max(a, b);
writeln( 'Max value is : ', ret );
end.
```

Când codul de mai sus este compilat și executat, acesta produce următorul rezultat-
Max value is : 200

Bibliografie

*manual pentru clasa 11-a la informatică,
Anatol Gremalschi

*https://www.tutorialspoint.com/pascal/pascal_functions.htm