

1. Competențe . . . . .	3
2. Efectul droste . . . . .	5
3. Noțiuni preliminare. . . . .	12
4. Mecanismul recursivității . . . . .	15
5. Exemple demonstrative . . . . .	18
6. Aplicații . . . . .	20
7. Bibliografie și webografie . . . . .	21



## Competențe generale

- *implementarea algoritmilor într-un limbaj de programare*
- *elaborarea algoritmilor de rezolvare a problemelor*
- *aplicarea algoritmilor fundamentali în prelucrarea datelor*
- *identificarea conexiunilor dintre informatică și societate*

## Competențe specifice

- *utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator*
- *construirea unor subprograme pentru rezolvarea subproblemelor unei probleme*
- *aplicarea mecanismului recursivității prin crearea unor subprograme recursive (definite de utilizator)*
- *compararea dintre implementarea recursivă și cea iterativă a aceluiași algoritm*
- *analiza problemei în scopul identificării subproblemelor acesteia*
- *descrierea metodei de rezolvare a unei probleme în termeni recursivi*



## Recursivitate. Funcții recursive



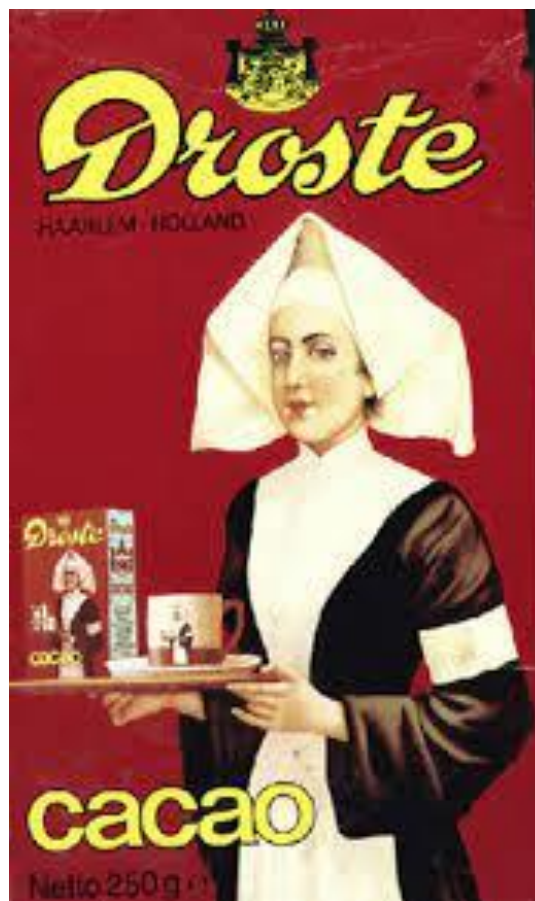
## 2. Efectul droste

- **Efectul Droste**, reprezintă efectul prin care o imagine apare în mod recursiv în interiorul său, de obicei în locuri în care o cu totul altă imagine ar fi de așteptat să apară.
- Efectul a fost denumit după o marcă olandeză de cacao, a cărei ilustrație de pe cutia de depozitare a fost realizată de Jan Misset în 1904.
- Încă din Evul Mediu acest efect a fost utilizat într-o oarecare măsură în cadrul unor opere de artă precum *Tripticul lui Stefaneschi* realizat de Giotto di Bondone în 1320. Unele cărți medievale repetă recursiv propria lor imagine, iar vitraliile unor biserici prezintă copii miniaturale ale aceluiași vitraliu.



- **Efectul Droste. Origine**

Efectul a fost denumit după imaginea de pe cutiile în care era ambalată cacaoa Droste, una dintre marile companii olandeze, imagine ce înfățișa o dădacă purtând o tavă pe care se află o ceașcă și un recipient de cacao cu aceeași imagine, concepută de Jan Misset



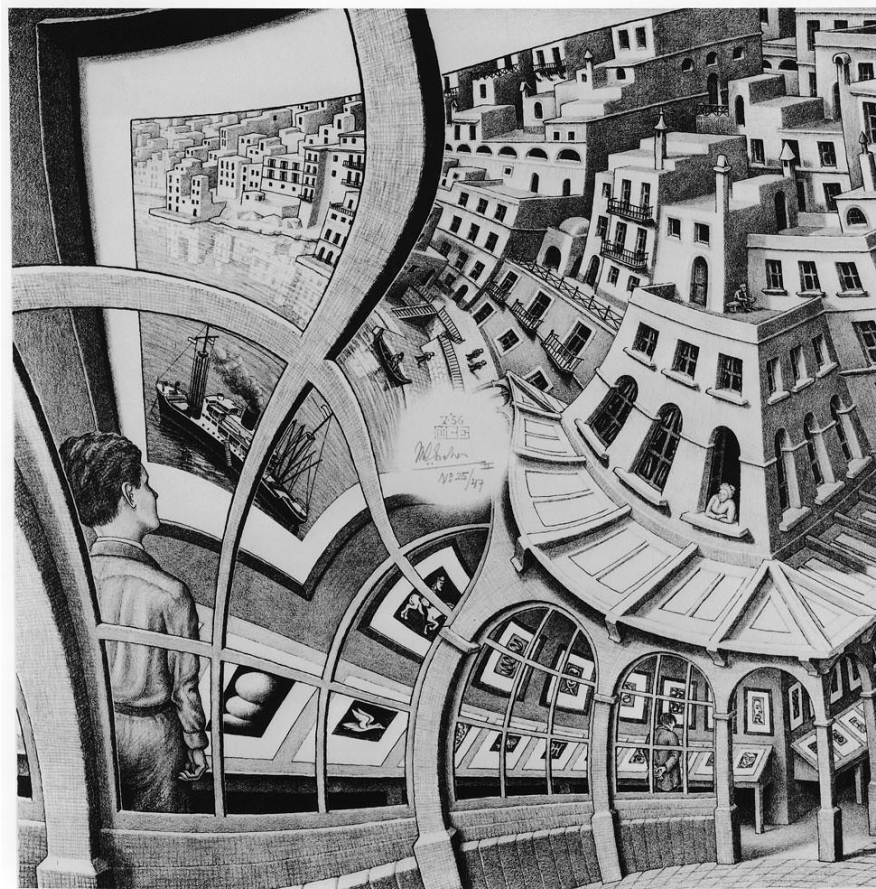
- **Efectul Droste. În arta medievală**

Efectul a fost utilizat de Giotto în 1320, în momentul în care a realizat Tripticul lui Stefaneschi. Polipticul îl prezintă în piesa centrală pe Cardinalul Giacomo Gaetani Stefaneschi oferind tripticul Sfântului Petru. Obiectul de artă a fost comandat de cardinal pentru unul dintre altarele Vechii Basilici Sf. Petru din Roma și se află actualmente la galeria de artă Pinacoteca Vaticana.



- **Efectul Droste.** În arta lui Maurits Cornelis Escher

Artistul olandez M. C. Escher a folosit efectul Droste în litografia „Prententoonstelling” publicată în 1956 care prezintă o galerie ce conține o tipăritură care înfățișează aceeași galerie, de fiecare dată imaginea fiind micșorată și rotită, cu excepția unui spațiu gol situat în centrul imaginii.



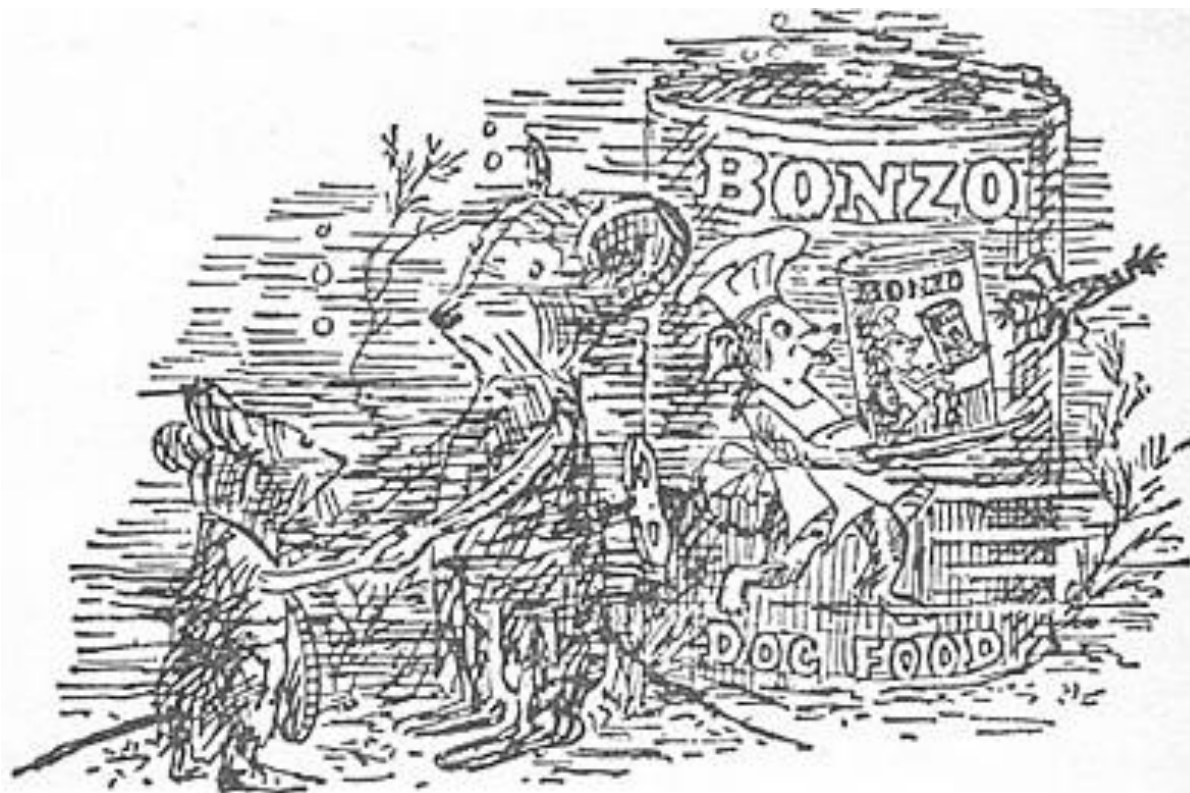
- **Efectul Droste. Utilizări moderne**

Efectul Droste a fost folosit în cadrul ambalajelor untului Land O'Lakes, imaginea prezentând o amerindiană ținând un pachet de unt ce are imaginea ei pe ambalaj.



- **Efectul Droste. Utilizări moderne**

Efectul Droste reprezintă o temă întâlnită și în poveștile pentru copii scrise de Russell Hoban, cum ar fi „*The Mouse and His Child*” în care este prezent sub forma etichetei unei cutii de mâncare pentru câini care conține o imagine a unei alte cutii de același fel.



- **Efectul Droste. Utilizări moderne**

Coperta albumului Ummagumma din 1969 al formației Pink Floyd înfățișează un membru al trupei șezând, pe perete existând o imagine ce conține un alt membru al formației. Acest lucru se repetă pentru toți cei patru membrii, ultima imagine fiind coperta albumului A Saucerful of Secrets.



## Recursivitate

Recursivitatea este un mecanism general de scriere a problemelor.

Un algoritm recursiv se caracterizează prin proprietatea că se autoapelează, adică din interiorul lui se apelează pe el însuși.

Atunci când se scrie un algoritm recursiv este suficient să se gândească ce se întâmplă la un anumit nivel, pentru că la orice nivel se întâmplă exact același lucru.

Într-un algoritm recursiv, pentru a realiza un anumit calcul sunt necesare două elemente:

1. o formulă de recurență;
2. o valoare inițială cunoscută.



Un exemplu de recursivitate este în definiția formală a numerelor naturale din cadrul teoriei mulțimilor:

- baza recursiei este faptul că 1 este număr natural;
- în plus, orice număr natural are un succesor, care este de asemenea un număr natural.

Un alt exemplu ar fi definiția conceptului de strămoș al unei persoane:

- un părinte este strămoșul copilului (“baza”);
- părinții unui strămoș sunt și ei strămoși (“pasul de recursie”).

- recurență = repetiție, revenire;
- formulă de recurență = formulă care exprimă orice termen dintr-un șir, în funcție de termenii precedenți;



**Exemplu**

Să se calculeze suma primelor  $n$  numere naturale.

$$S(n) = 1 + 2 + 3 + \dots + (n-1) + n$$

Formula matematică:

$$S(n) = S(n-1) + n$$

// formula de recurență

$$S(0) = 0$$

//valoarea inițială cunoscută

$$S(n) = \begin{cases} 0, & \text{pentru } n = 0 \\ S(n-1) + n, & \text{pentru } n > 0 \end{cases}$$

$$S(4) = 1 + 2 + 3 + 4 = 10$$

astfel:

$$S(4) = S(3) + 4$$

$$S(0) = 0$$

$$S(3) = S(2) + 3$$

$$S(1) = S(0) + 1 = 0 + 1 = 1$$

$$S(2) = S(1) + 2$$

$\Rightarrow$

$$S(2) = S(1) + 2 = 1 + 2 = 3$$

$$S(1) = S(0) + 1$$

$$S(3) = S(2) + 3 = 3 + 3 = 6$$

$$S(0) = 0$$

$$S(4) = S(3) + 4 = 6 + 4 = 10$$



## 4. Mecanismul recursivității

Un algoritm recursiv se implementează folosind o funcție recursivă.

Se numește **funcție recursivă** o funcție care din corpul ei se apelează pe ea însăși. Un subprogram se numește **recursiv** dacă se autoapelează.

Orice funcție recursivă trebuie să îndeplinească două condiții:

1. să se poată executa cel puțin o dată fără să se autoapeleze;
2. toate apelurile să se producă astfel încât să se tindă spre îndeplinirea condiției de execuție fără apeluri.

Din afara funcției recursive, se face un prim apel la funcția recursivă, după care funcția se autoapeleză de un anumit număr de ori.

Pentru orice algoritm iterativ există un algoritm recursiv echivalent (rezolvă aceeași problemă) și invers, pentru orice algoritm recursiv există unul iterativ echivalent.



## Exemplu

Să se calculeze suma primelor  $n$  numere naturale.

$$S(n) = 1 + 2 + 3 + \dots + (n-1) + n$$

```
1  #include <iostream>
2
3  using namespace std;
4
5  int S(int n)
6  {
7      if(n==0)
8          return 0;
9      else
10         return S(n-1)+n;
11 }
12
13 int main()
14 {
15     cout <<S(4)<< endl;
16     return 0;
17 }
18
```



Autoapelarea se poate realiza în două moduri:

1. *direct* – în corpul funcției apare explicit un apel recursiv;
2. *indirect* – în corpul funcției apare apelul unei alte funcții care, la rândul său, apelează direct sau indirect funcția recursivă.

Recursivitate directă	Recursivitate indirectă
<pre>void A() {     ...     A();     ... }</pre> <pre>void main() {     ...     A();     ... }</pre>	<pre>void A() {     ...     B();     ... }</pre> <pre>void B() {     ...     A();     ... }</pre> <pre>void main() {     ...     A();     ... }</pre>

### Exemple demonstrative

1. Pentru subprogramul `f` definit mai jos, ce se afișează ca urmare a apelului `f(3,17)`?

```
void f ( int a, int b)
{ if(a<=b)
  {   cout<<'*';
      f(a+1,b-2);
  }
  else cout<<b;
}
```

2. Pentru subprogramul `f` definit mai jos, ce se afișează ca urmare a apelului `f(3,17)`?

```
void f ( int a, int b)
{ if(a<=b)
  {   f(a+1,b-2);
      cout<<'*';
  }
  else cout<<b;
}
```

3. Pentru subprogramul `f` definit mai jos, ce se afișează ca urmare a apelului `f(3,17)`?

```
void f ( int a, int b)
{ if(a<=b)
  {   f(a+1,b-2);
      cout<<'*';
  }
  cout<<b;
}
```



4. Ce valoare are  $f(4063)$  pentru funcția  $f$ , definită alăturat?

```
long f(long n)
{
    if (n==0) return 1;
    else
        if (n%10==0)
            return f(n/10);
        else
            return (n%10) * f(n/10);
}
```



### Fișă de lucru

- Subprograme recursive recursive
- Aplicații subprograme recursive



## 7. Bibliografie și webografie

1. Miloșescu M., *Informatica. Manual pentru clasa a X*, Editura Didactică și Pedagogică, București, 2005
2. Mateescu G, Moraru P., *Informatica. Manual pentru clasa a X*, Editura Donaris, Sibiu, 2006
3. Popescu C., *Culegere de probleme de informatică*, Editura Donaris-Info, Sibiu, 2002
4. Ministerul Educației, Cercetării și Tineretului, Centrul Național pentru Curriculum și Evaluare în Învățământul Preuniversitar, *Proba scrisă la informatică. Examenul de bacalaureat – Variante (1-100)*, București 2008
5. <http://ro.wikipedia.org/wiki/Recursivitate>
6. <http://en.wikipedia.org/wiki/Recursion>
7. [https://ro.wikipedia.org/wiki/Efectul\\_Droste](https://ro.wikipedia.org/wiki/Efectul_Droste)

