

# STRUCTURA REPETITIVĂ

## AMINTIȚI-VĂ

Ce este un algoritm?

Cu ce lucrează algoritmii?

Ce tipuri de date cunoașteți?

Fie următoarea problemă:

Diriginta unei clase dorește să cumpere  $n$  cărți pentru fiecare elev al clasei, pentru ca aceștea să citească în vacanța de vară. S-a hotărât să nu cumpere cărți identice tuturor elevilor, ci personalizat, în funcție de pasiunile fiecăruia. Realizați un algoritm cu ajutorul căruia puteți determina cât a plătit doamna dirigintă pe cărți.

## OBSERVAȚI

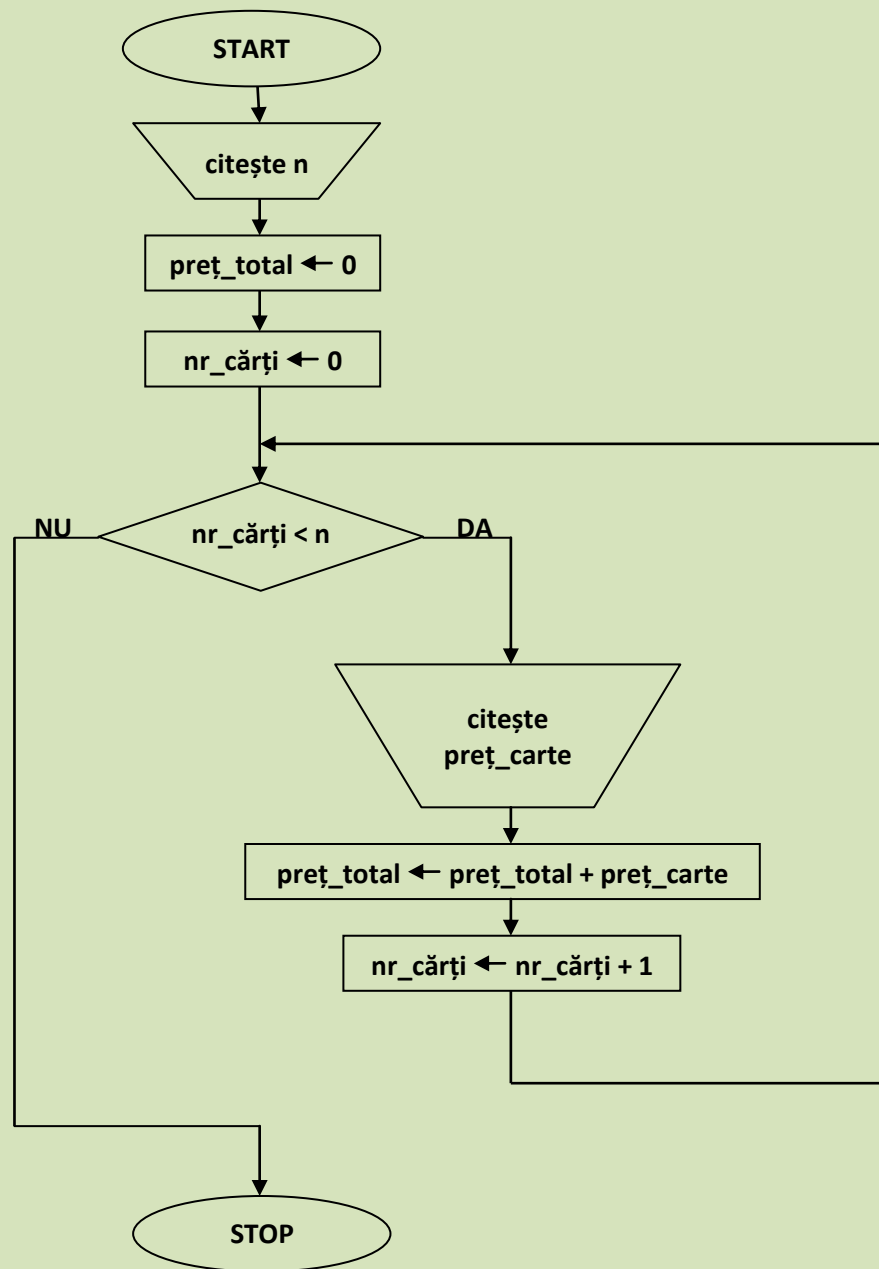
Va trebui să adunăm pe rând prețul fiecărei cărți în parte, până când ajungem la numărul de cărți pe care le dorim (care este egal cu numărul de elevi ai clasei).

Pentru asta avem nevoie de 4 variabile:

- $n$  – numărul elevilor din clasă;
- $nr\_cărți$  – numărul cărților cumpărate la un moment dat;
- $preț\_carte$  – prețul cărții curente (de exemplu dacă  $nr\_cărți = 7$  vom introduce prețul celei de a 7-a carte);
- $preț\_total$  – valoarea totală plătită la un moment dat;

Observăm că există niște pași care se tot repetă. Dacă la început avem variabila  $preț\_total = 0$  (deoarece nu am cumpărat nici o carte) și variabila  $nr\_cărți = 0$ , după primul pas, va trebui să avem  $nr\_cărți = 1$  și  $preț\_total = \text{valoarea primei cărți}$ ; după cel de-al 2-lea pas vom avea  $nr\_cărți = 2$  și  $preț\_total = \text{valoarea primei cărți} + \text{valoarea celei de a 2-a cărți}$ , și așa mai departe. La fiecare pas în parte trebuie să citim valoarea cărții și să o atribuim variabilei  $preț\_carte$ .

Pentru a putea crea algoritmul, vom realiza pentru început schema logică:



Putem observa că există instrucțiuni care se repetă de mai multe ori (în acest caz, se repetă de atâtea ori câți elevi sunt în clasă). Dacă, să presupunem că în clasă sunt 30 de elevi, ar exista 2 instrucțiuni (*citește preț\_carte* și *preț\_total ← preț\_total + preț\_carte*) pe care ar trebui să le scriem de 30 de ori. Pentru a simplifica algoritmul, ne vine în ajutor structura repetitivă.

## REȚINEȚI

Structura repetitivă, se utilizează pentru efectuarea în mod repetat a unui set de instrucțiuni. Repetarea poate fi condiționată la începutul sau la finalul structurii, sau poate fi exact definită. Structura repetitivă se mai numește buclă.

Există trei tipuri de structuri repetitive:

- cu număr necunoscut de pași;
  - o repetitivă cu test inițial (condiționată anterior);
  - o repetitivă cu test final (condiționată posterior);
- cu număr cunoscut de pași;
  - o repetitivă cu număr cunoscut de pași (cu contor);

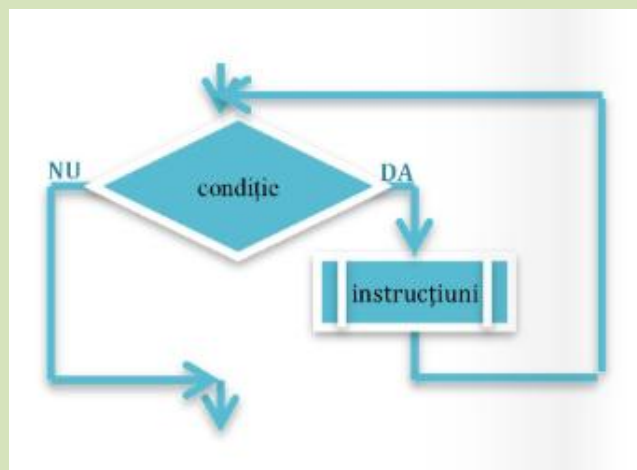
### Structura repetitivă cu test inițial (condiționată anterior)

Structura repetitivă cu test inițial este o structură care are condiția aflată la **începutul** structurii.

Aceasta are forma generală dată de **sinatxa** în limbaj pseudocod:

**cât timp (condiție) execută**  
**< secvență de instrucțiuni >**  
**sfârșit cât timp**

**Schema logică** a acesteia este:



**Modul de execuție** al structurii repetitive cu test inițial este următorul:

- se verifică dacă este sau nu îndeplinită condiția;
- se evaluează valoarea de adevăr a condiției:
  - o când condiția este îndeplinită, se execută setul de instrucțiuni care se află pe ramura DA, iar apoi se revine la blocul decizional (la condiție);
  - o când condiția nu este îndeplinită, se iese din blocul decizional;
- pentru a se executa cel puțin o dată setul de instrucțiuni din cadrul structurii, condiția trebuie să fie îndeplinită la prima executare a structurii.

Pentru a se evita intrarea într-un ciclu infinit (adică pentru a putea ajunge la final), este necesar ca în secvența de instrucțiuni să se modifice cel puțin o variabilă care se află în condiție, pentru ca la un moment dat condiția să devină falsă.

## Structura repetitivă cu test final (condiționată posterior)

Structura repetitivă cu test final sau condiționată posterior este o structură care are condiția aflată la **finalul** structurii.

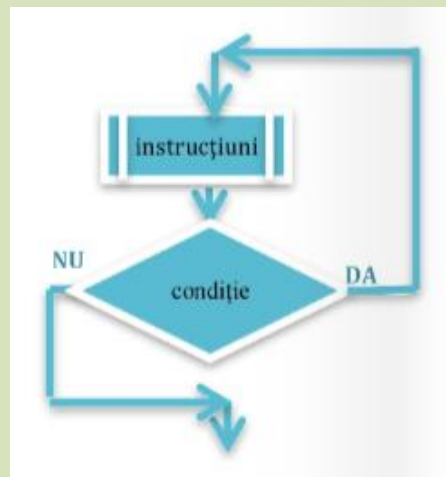
Structura repetitivă cu test final se regăsește sub două forme:

1. Structura care repetă setul de instrucțiuni **cât timp condiția este adevărată**;

Aceasta are forma generală dată de **sinatxa** în limbaj pseudocod:

```
┌ execută  
│ < secvență de instrucțiuni >  
└ cât timp (condiție)
```

**Schema logică** a acesteia este:



**Modul de execuție** al structurii repetitive cu test inițial este următorul:

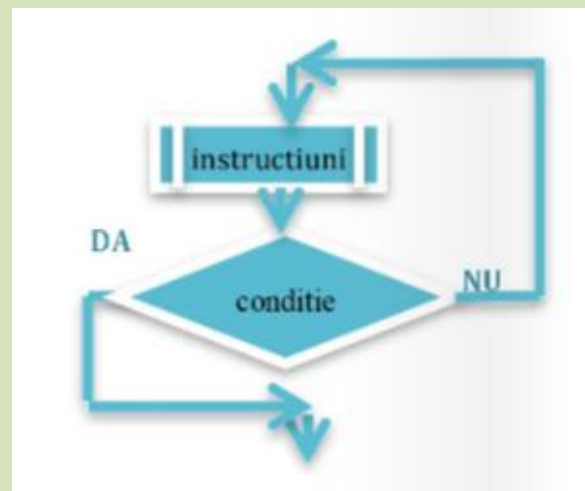
- se execută setul de instrucțiuni;
- se evaluează condiția:
  - o când condiția este îndeplinită, se reia execuția setului de instrucțiuni care se află pe ramura DA;
  - o când condiția nu este îndeplinită, se iese din structura repetitivă;
- setul de instrucțiuni din cadrul structurii se execută cel puțin o dată (indiferent dacă este sau nu îndeplinită condiția, deoarece condiția se află la sfârșit, după executarea setului de instrucțiuni).

2. Structura care repetă setul de instrucțiuni **până când** condiția este adevărată;

Aceasta are forma generală dată de **sinatxa** în limbaj pseudocod:

```
[ repetă  
  < secvență de instrucțiuni >  
 până când (condiție)
```

**Schema logică** a acesteia este:



**Modul de execuție** al structurii repetitive cu test inițial este următorul:

- se execută setul de instrucțiuni;
- se evaluează condiția:
  - o când condiția este falsă, se reia execuția setului de instrucțiuni care se află pe ramura NU;
  - o când condiția este îndeplinită, se iese din structura repetitivă;
- setul de instrucțiuni din cadrul structurii se execută cel puțin o dată (indiferent dacă este sau nu îndeplinită condiția, deoarece condiția se află la sfârșit, după executarea setului de instrucțiuni).

Diferența dintre structura repetitivă cu test final și structura repetitivă cu test inițial este aceea că, în cazul primeia, instrucțiunile din interiorul structurii se execută cel puțin o dată, iar, în cazul celei de a doua, instrucțiunile se execută doar dacă este adevărată condiția.

### OBSERVAȚI

Fie următoarea problemă:

*Se introduc de la tastatură numere, până la introducerea numărului 0. Stabiliți câte numere pozitive și câte numere negative au fost introduse de la tastatură.*

Vom urmări rezolvarea problemei folosind cele 3 tipuri de structuri repetitive:

a) Folosind structura repetitivă cu test inițial

**început algoritm**

întreg  $n$ , *pozitiv*, *negativ*;

*pozitiv* ← 0;

*negativ* ← 0;

cât timp ( $n \neq 0$ ) execută

    dacă ( $n > 0$ ) atunci

*pozitiv* ← *pozitiv* + 1;

    altfel

*negativ* ← *negativ* + 1;

    sfârșit dacă

        citește  $n$ ;

sfârșit cât timp

scrie *pozitiv*;

scrie *negativ*;

**sfârșit algoritm**

b) Folosind structura repetitivă cu test final de tip *cât timp*

**început algoritm**

întreg  $n$ ,  $pozitiv$ ,  $negativ$ ;

$pozitiv \leftarrow 0$ ;

$negativ \leftarrow 0$ ;

```
execută
    citește  $n$ ;
    dacă ( $n > 0$ ) atunci
         $pozitiv \leftarrow pozitiv + 1$ ;
    sfârșit dacă
    dacă ( $n < 0$ ) atunci
         $negativ \leftarrow negativ + 1$ ;
    sfârșit dacă
cât timp ( $n <> 0$ )
    scrie  $pozitiv$ ;
    scrie  $negativ$ ;
sfârșit algoritm
```

c) Folosind structura repetitivă cu test final de tip *până când*

**început algoritm**

întreg  $n$ ,  $pozitiv$ ,  $negativ$ ;

$pozitiv \leftarrow 0$ ;

$negativ \leftarrow 0$ ;

```
repetă
    citește  $n$ ;
    dacă ( $n > 0$ ) atunci
         $pozitiv \leftarrow pozitiv + 1$ ;
    sfârșit dacă
    dacă ( $n < 0$ ) atunci
         $negativ \leftarrow negativ + 1$ ;
    sfârșit dacă
până când ( $n = 0$ )
    scrie  $pozitiv$ ;
    scrie  $negativ$ ;
sfârșit algoritm
```

## APLICAȚIE

Scrieți algoritmul pentru determinarea parității unui număr introdus de la tastatură, până la introducerea numărului 0, folosind toate tipurile de structuri repetitive cunoscute. În cazul în care numărul introdus este par, algoritmul afișează textul „număr par”, iar în caz contrar, algoritmul afișează textul „număr impar”.