

STRUCTURI DE CONTROL

Algoritm - succesiune de etape de calcul ce se poate aplica pentru rezolvarea unei clase de probleme.

Descrierea unui algoritm poate fi efectuată utilizând:

Scheme logice

Pseudocod

Obiecte cu care lucrează algoritmii

Date

Principalele obiecte ale unui algoritm sunt datele. Ele pot fi: Date

de intrare - care sunt cunoscute

Date de ieșire - rezultate furnizate

După tipul lor, datele pot fi:

Întregi: 2, -4

Reale: 3.25, 0.007

Logice: true și false – adevărat și fals

Caracter: „y”, „a”

Șir de caractere: “ab23_c”

Constante

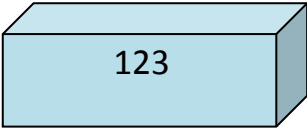
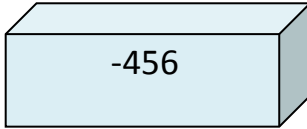
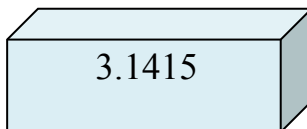
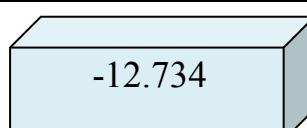
În descrierea unui algoritm pot apare **constantele**; acestea sunt date conținute în program care nu sunt citite sau calculate. Un exemplu este constanta π din matematică.

Variabile

Programele, și implicit algoritmii, lucreaza cu date. O **variabilă** este utilizată pentru a stoca (a păstra) o dată. Se numește **variabilă** pentru că valoarea stocată se poate schimba pe parcursul execuției algoritmului. O variabilă are un nume unic și un conținut care poate să difere de la un moment la altul al execuției algoritmului. Mai precis, o variabilă este o locație de memorie care are un nume și care păstrează o valoare de un anumit tip.

O variabilă este caracterizată prin:

- Nume
- Tip
- Valoarea la un moment dat
- Locul în memorie (adresa)

Nume	Valoare	Tip
Numar		Intreg
Suma		Intreg
Pi		Real
Medie		Real

Orice variabilă are un nume, conține o valoare declarată de un anumit tip, valoare memorată mereu la o aceeași adresă de memorie

Expresii

1. Aritmetice --- $2*(23/2+23\%2)\%2$
2. Logice --- $!(2+3>5)\&\&(4>3)$

Un algoritm poate efectua operații de:

Intrare(CITIRE): preluarea unei date de la un dispozitiv de intrare (consola, suport magnetic, etc)

Ieșire(SCRIERE): transferul unei date din memoria internă către un dispozitiv de ieșire

(ecran, imprimanta, suport magnetic, etc) **Atribuire(ATRIBUIRE):**

$x=3; y=x; y=x+y$ (variabila ia valoarea(=) expresiei)

- Operația de atribuire se realizează astfel:
 - Se evaluează expresia din partea dreapta a semnului =
 - Valoarea obținută este atribuită variabilei din stânga (stocată în locația sa de memorie), care își pierde vechea valoare

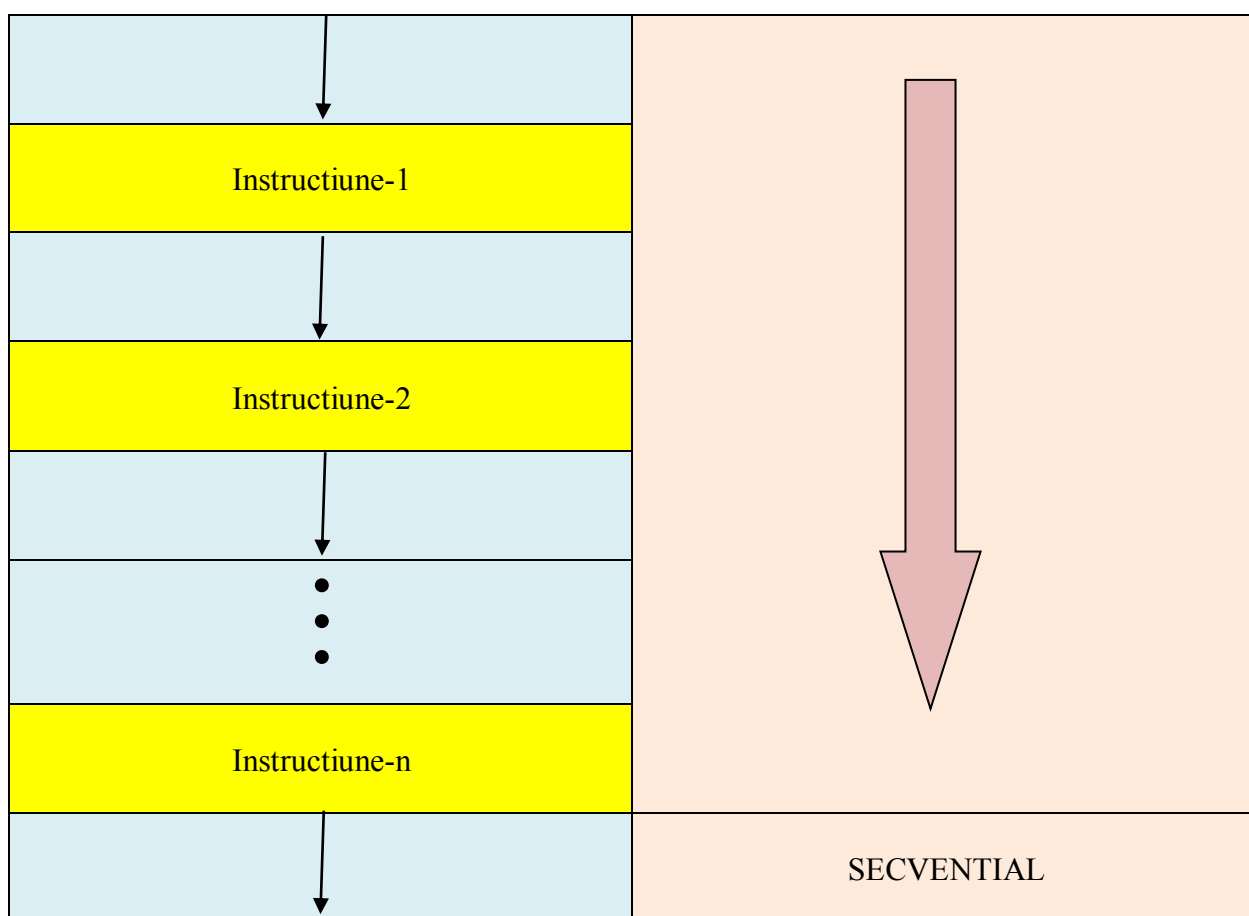
Decizie(DECIZIE): o întrebare ridicată de programator la un moment dat în program, operație prin care, în funcție de valoarea curentă a unei condiții, se alege următoarea etapă a algoritmului

ORDINEA EFECTUĂRII INSTRUCȚIUNILOR INTR-UN ALGORITM MAI POARTA DENUMIREA DE **STRUCTURA**

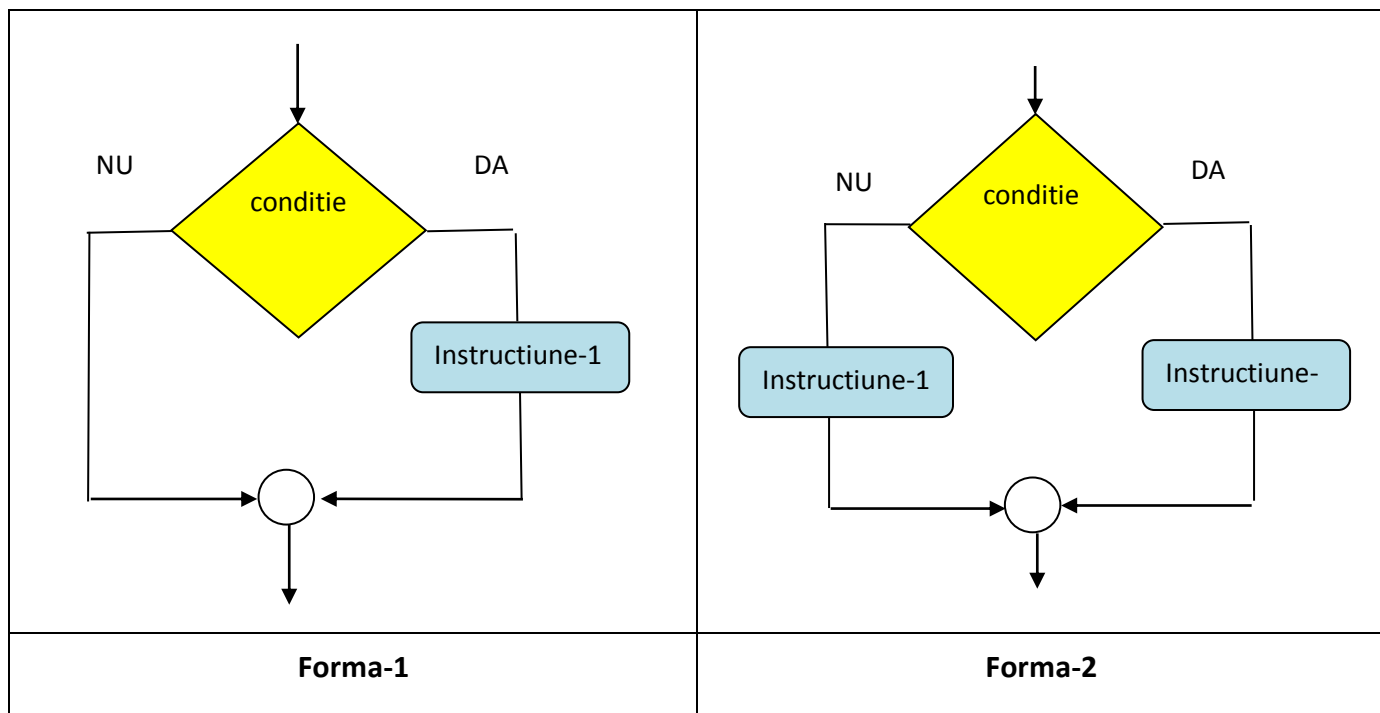
Orice algoritm poate fi compus din numai trei structuri de calcul:

1. structura secvențială(liniara) - secvența;
2. structura alternativă - decizia;
3. structura repetitivă - ciclul.
 - 3.1 Cu condiție inițială
 - 3.2 Cu condiție finală
 - 3.3 Cu contor

1. Secvența(structura liniară) se reprezintă prin simboluri grafice conectate prin săgeți ce sugerează fluxul operațiilor (secvențial):



Se execută în ordine *Instrucțiune-1*, apoi *Instrucțiune-2*, ș.a.m.d. până la *Instrucțiune-n*

1. Decizia(structura alternative) poate avea una din următoarele forme:**Forma 1:**

Se evaluează Condiție;

Dacă valoarea Condiției este adevărat, atunci se execută Blocul DA;

Dacă valoarea Condiției este fals, executia se continuă cu operația care urmează imediat după blocul DA.

Forma 2:

Se evaluează Condiție;

Dacă valoarea Condiției este adevărat se execută Blocul DA;

Dacă valoarea Condiției este fals se execută Blocul NU.

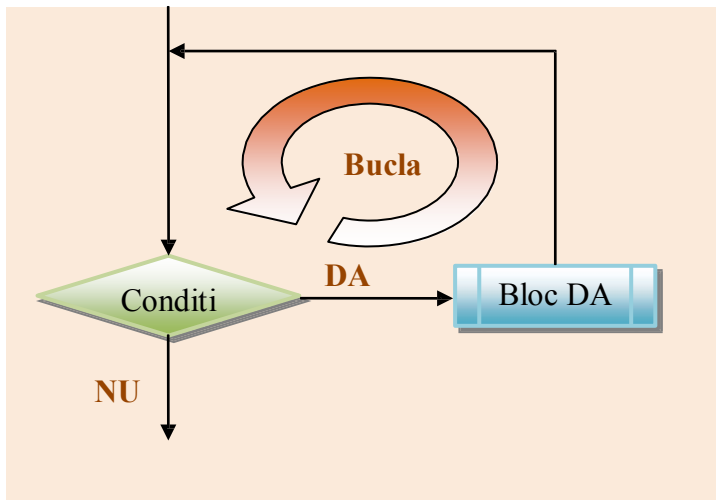
.1 Structura repetitivă cu condiție inițială (structură repetitivă de tip *while*)

Pas 1 : se evaluează *Condiție* ;

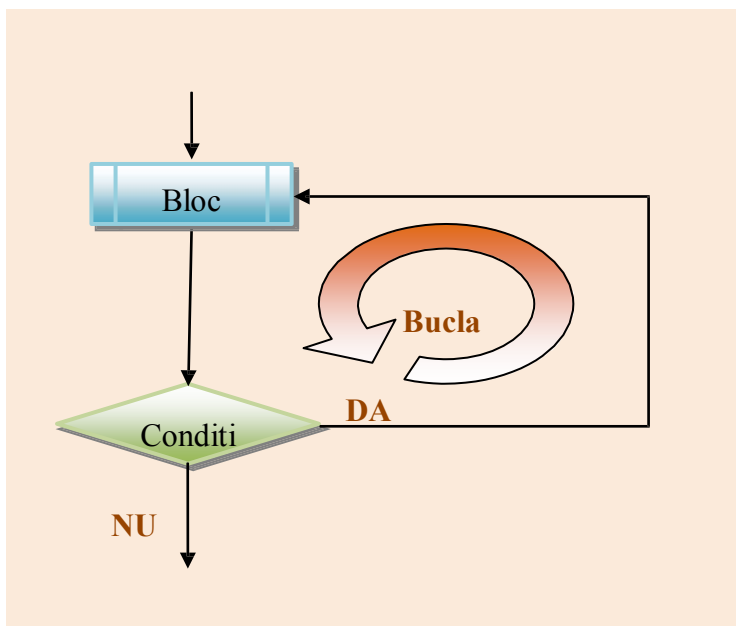
Pas 2:

dacă valoarea *Condiției* este fals (NU), se iese din structura repetitivă;

**dacă valoarea expresiei este adevărat (DA), se execută Bloc DA ,
apoi se reia execuția de la Pas 1**



3.2 Structura repetitivă cu condiție finală (structură repetitivă de tip *do-while*)

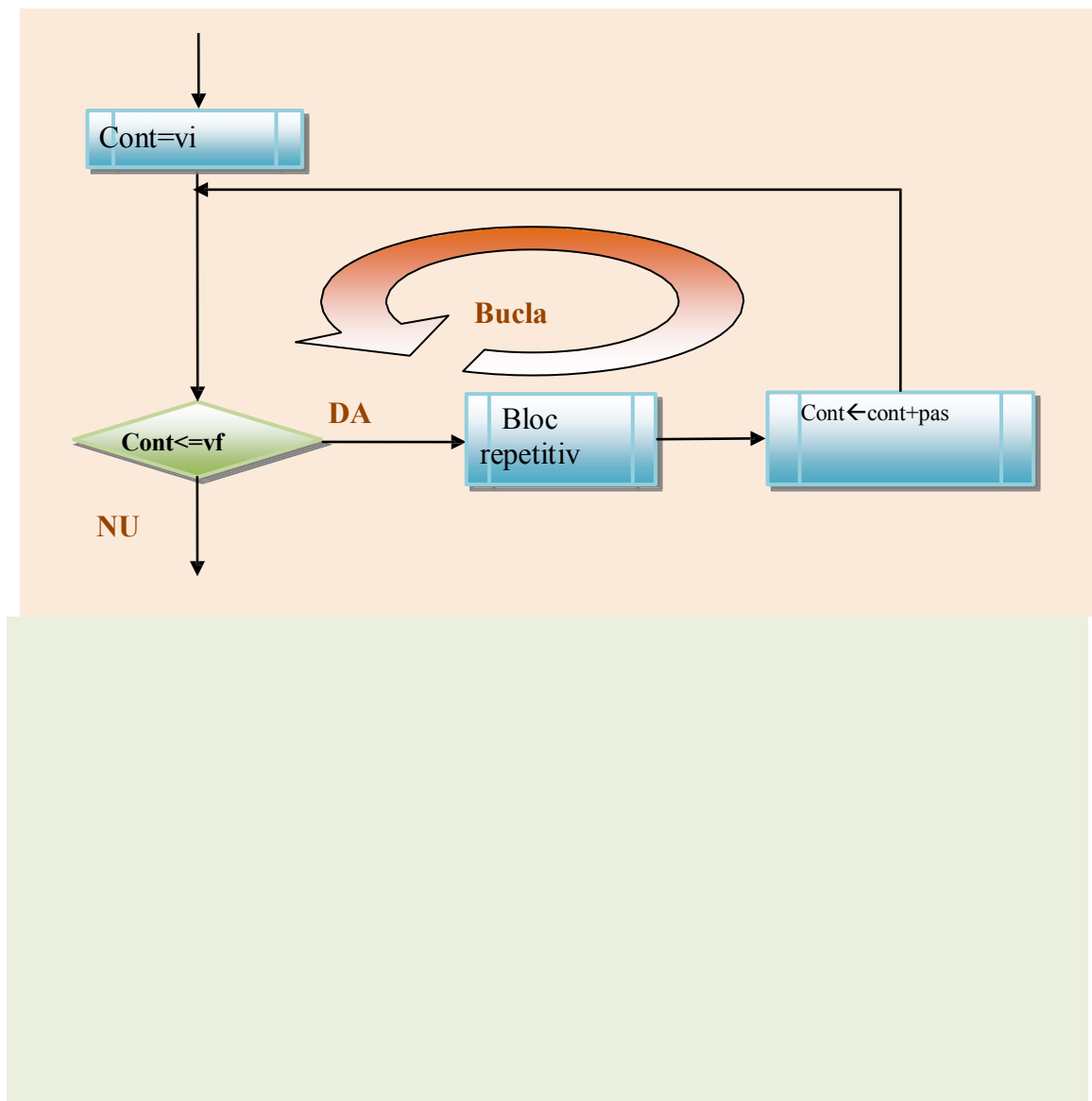


Pas 1 : se execută instrucțiunea sau instrucțiunile din *Bloc*;

Pas 2: se evaluează *Condiție*;

Pas 3: dacă valoarea *Condiției* este fals (NU), se iese din instrucțiunea repetitivă;
dacă valoarea expresiei este adevărat (DA) se reia de la Pas 1

3.3 Structura repetitivă cu contor (structură repetitivă de tip *for*)



Pas 1: se initializeaza variabila contor cu valoarea initiala vi.

Pas 2: se evaluează *conditia*

cont<=vf;

- Dacă valoarea expresiei *cont<=vf* este fals, atunci se iese din structura repetitivă.
- dacă valoarea expresiei *cont<=vf* este adevărat, atunci :
 - se execută *Blocul repetitiv*
 - se execută blocul *,se mareste contor cu valoarea pas si se merge la Pas 2.*

REPREZENTAREA STRUCTURILOR IN PSEUDOCOD

Structura de decizie

Forma 1

Daca conditie atunci

Bloc DA

Sfarsit daca

Forma 2

Daca conditie atunci

Bloc DA

Altfel

Bloc NU

Sfarsit daca

Structura repetitiva cu test initial(while)

Cat timp conditie executa

Bloc instructiuni

Sfarsit cat timp

Structura repetitiva cu test final(do-while)

executa

Bloc instructiuni

Cat timp contitie

Structura repetitiva cu contor(for)

Pentru $i \leftarrow v_i, v_f, pas$, executa

Bloc instructiuni

Sfarsit conditie