

Reprezentarea algoritmilor

Scheme logice

Variante de reprezentare a algoritmilor

- Există numeroase variante de reprezentare a algoritmilor;
 - În general cu cât acestea sunt mai sugestive pentru oameni, cu atât sunt mai greu de înțeles și interpretat pentru sistemele de calcul și invers;
 - Cele mai frecvent folosite metode reprezentare sunt: schemele logice, pseudocodul, limbajele de programare de nivel înalt (Pascal, C, C++, Java, C#, Python etc.), limbajele de asamblare, limbajele cod-mașină.
-

Schemele logice

- Sunt desene care respectă niște reguli stricte, fiind formate dintr-un număr de blocuri, conectate între ele, cu un singur punct de pornire (START) și un unic punct final (STOP);
 - Blocurile din care e alcătuită o schemă logică pot fi încadrate în 4 categorii:
 - Blocuri START / STOP;
 - Blocuri pentru operații de INTRARE / IEȘIRE;
 - Blocuri de atribuire;
 - Blocuri de decizie.
-

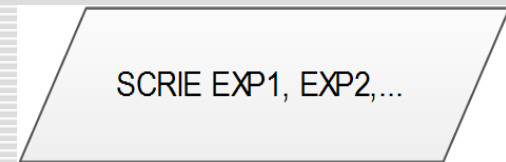
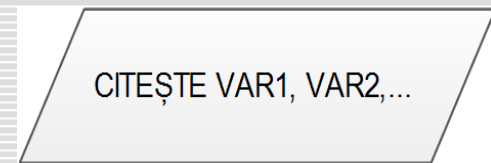
Blocurile START / STOP

- ❑ Fiecare schemă logică trebuie să aibă exact un bloc START și unul STOP ;
- ❑ Ambele au o formă ovală ca în figura alăturată.



Blocurile de INTRARE / IEȘIRE

- În general algoritmi preiau date din exterior (date de intrare), pe care le prelucreează, furnizând în final rezultate (date de ieșire) ;
- Preluarea datelor de intrare se face prin operații de “citire”, iar transmiterea rezultatelor (date de ieșire) prin operații de “afișare” ;
- Pentru ambele tipuri de operații se folosesc blocuri în formă de paralelogram ca în figurile alăturate.



Blocurile de atribuire

- Corespund prelucrărilor efectuate în cadrul algoritmului;
- Blocurile sunt dreptunghiuri ca în figura alăturată.

VARIABILĂ ← EXPRESIE

Despre variabile

- ❑ Prelucrarea datelor presupune stocarea acestora;
 - ❑ “Containerele” în care se păstrează datele poartă numele de variabile și ni le putem imagina ca pe niște cutii în care la un moment dat putem depozita un singur obiect;
 - ❑ Variabilele sunt folosite atât pentru păstrarea datelor de intrare, cât și pentru reținerea unor rezultate (parțiale sau finale) obținute pe parcursul derulării algoritmului;
 - ❑ Fiecare operație de citire presupune și specificarea variabilelor în care vor fi păstrate datele;
 - ❑ Fiecare atribuire are forma: **VARIABILĂ ← EXPRESIE** (se citește “VARIABILĂ IA VALOAREA EXPRESIE”).
-

Despre expresii

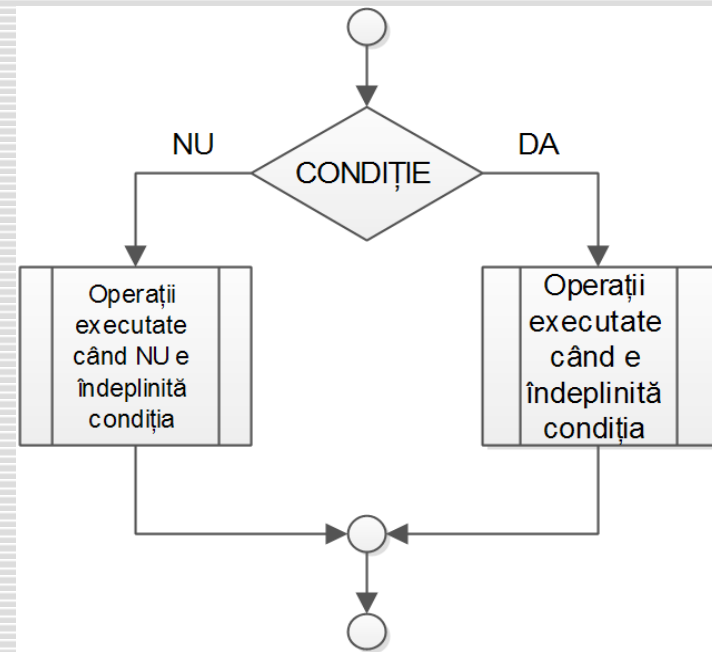
- Expresiile sunt formate din operatori (+, -, *, ?, % etc.) și operanzi;
 - Operanzii pot fi variabile, constante sau alte expresii (plasate între paranteze);
 - Exemple: $2*(5+x)$: operatorii sunt * și +; pentru operatorul + operanzii sunt 5 (constantă) și x (variabilă), iar pentru operatorul * operanzii sunt 2 și $5+x$ (expresie).
-

Blocuri de decizie

- Apar acolo unde algoritmul se ramifică în funcție de verificarea unei condiții (expresie logică, având valorile ADEVĂRAT sau FALS);
 - Întotdeauna ramura din dreapta va corespunde valorii “ADEVĂRAT” (“DA”, condiția este îndeplinită), iar cea din stânga valorii “FALS” (“NU”, condiția NU este îndeplinită);
 - Blocurile de decizie se folosesc și în cazul în care se dorește ca anumite operații să se repete cât timp o anumită condiție e îndeplinită; în astfel de situații, ramura “DA” se va întoarce în schemă undeva deasupra blocului de decizie.
-

Blocuri de decizie simple

- Apar acolo unde nu este necesară repetarea anumitor operații;
- Sunt simbolizate prin romburi, în interiorul cărora apare câte o expresie logică (condiție).



Blocuri de decizie pentru structuri repetitive

- Apar atunci când anumite operații trebuie repetate în funcție de verificarea unei anumite condiții;
- Au tot forma unui romb, dar săgeata corespunzătoare ramurii “NU” revine undeva deasupra.

