

Structuri de control

Acest **curs** prezintă **Structuri de control**.

În acest PDF poți vizualiza cuprinsul și bibliografia (daca sunt disponibile) și aproximativ două pagini din documentul original.

Arhiva completa de pe site conține un fișier, într-un număr total de **11 pagini**.

Fișierele documentului original au următoarele extensii: doc.

Extras

Forme de reprezentare a algoritmilor:

1. Schema logică (organigramă)
2. Limbajul algoritmic (pseudocod)
3. Programul

Schemele logice sunt forme de reprezentare grafică a algoritmilor, folosind simboluri a căror formă indică tipul acțiunii. Aceste simboluri se numesc blocuri. În interiorul blocurilor, de la caz la caz, pot figura una sau mai multe operații care descriu efectiv acțiunea ce se execută. Ordinea de execuție a operațiilor este dată de segmentele și săgețile ce fac legătura dintre blocuri.

Tipuri de blocuri:

1. Blocurile de început și de sfârșit (blocuri terminale) se pun la începutul și, respectiv, la sfârșitul unei scheme logice. Blocurile terminale pot apărea o singură dată într-o schemă logică.
2. Blocurile de intrare/ ieșire (de citire/ scriere) pun în evidență informațiile de intrare (citire), respectiv ieșire (scriere).
3. Blocul de atribuire desemnează operații de calcul de tipul atribuirilor și are forma:
4. Blocul de decizie indică situațiile în care algoritmul se ramifică în funcție de anumite condiții (propoziții logice aritmetice). Dacă expresia logică este adevărată se merge pe ramura lui DA, iar dacă expresia logică este falsă se merge pe ramura lui NU.
5. Blocul conector pune în evidență nodurile (punctele de intersecție) ale unui algoritm. El se simbolizează printr-un cerc mic.
6. Blocul de continuare este folosit pentru a putea continua scrierea unei scheme logice pe o altă pagină.

Pentru întocmirea unei scheme logice corecte sunt necesare respectarea unor anumite reguli după cum urmează:

- orice schemă logică se întocmește de sus în jos;
- legătura între blocurile schemei se face numai prin săgeți, pentru a stabili sensul de parcurgere;
- în cazul întâlnirii unui bloc de decizie se precizează deasupra liniilor de legătură când se execută expresia

Limbajul pseudocod

Reprezentarea algoritmilor în pseudocod

Algoritmul este un concept abstract. Reprezentarea algoritmului înseamnă implementarea fizică a algoritmului. Algoritmul prin care se descrie o problemă care trebuie să fie rezolvată de calculator nu trebuie să fie ambiguu deoarece, în cazul rezolvărilor neclare, calculatorul nu poate să opteze singur pentru o anumită posibilitate. Pentru a evita ambiguitatea descrierii unui algoritm printr-un limbaj natural (limba în care vorbim), se poate folosi pentru reprezentarea lui un limbaj artificial numit pseudocod, apropiat de limbajul de programare, dar care este puțin formalizat și nu este constrâns de regulile de sintaxă ale limbajului de programare.

Pseudocodul este o forma de reprezentare a algoritmilor și ca orice limbaj folosește un grup de cuvinte cheie (citește, scrie, dacă, atunci, cât timp, pentru etc.), organizate în construcții sintactice care traduc structurile fundamentale, dar neavând o sintaxă rigidă ca un limbaj de programare. Limbajul pseudocod lasă la libera alegere a utilizatorului introducerea unor notații care să permită o mai bună exprimare.

Pseudocodul (codul fals) este considerat un cod fals deoarece nu poate fi folosit pentru a exprima instrucțiunile care se dau calculatorului pentru a rezolva problema descrisă de algoritm (nu poate fi folosit ca limbaj de programare). El folosește expresii, din limbajul natural, în care exprimarea acțiunilor care se execută se face prin propoziții care se termină prin simbolul punct și virgulă (;). În propoziții se folosesc cuvinte cheie, pentru descrierea structurilor de control și a operațiilor de comunicare. O propoziție care reprezintă un pas de comunicare sau de acțiune începe obligatoriu cu un verb.

În cazul pasului de comunicare, verbul este citește (read) pentru o operație de intrare și scrie (write) pentru o operație de ieșire. Verbul este urmat de lista datelor care se comunică. Lista conține date variabile reprezentate prin identificatorii lor. În cazul unei operații de scriere lista poate conține și date constante reprezentate prin valoare. Elementele listei se separă prin virgulă.

Cuvintele cheie pot fi în limba română (pseudocodul în limba română) sau în limba engleză (pseudocodul în limba engleză).

Pentru a delimita secvența de descriere a datelor de secvența de descriere a pașilor algoritmului, pașii algoritmului vor fi încadrați de cuvintele cheie început...sfârșit (begin...end).

.....
.....
.....

Documentul complet de 11 pagini il poti citi daca il descarci din Biblioteca.RegieLive.ro

Imagini din documentul complet:

Structuri de bază

Tipuri de structură

1. Structura liniară
2. Structura decizională alternativă
 - 2.1. Structura decizională simplă
 - 2.1.1. Structura decizională completă
 - 2.1.2. Structura decizională cu o ramură vidă
 - 2.2. Structura de selecție condiționată (structura alternativă condiționată)
3. Structura repetitivă
 - 3.1. Structura repetitivă cu număr necunoscut de pași
 - 3.1.1. Structura repetitivă cu tot întregul condițional anterior, WHILE-DO
 - 3.1.2. Structura repetitivă cu tot final condițional posterior, REPET-UNTIL
 - 3.2. Structura repetitivă cu număr cunoscut de pași
 - 3.2.1. Structura repetitivă cu conținut (cu număr cunoscut de pași, FOR)

1. Structura liniară

Structura caracterizată prin o succesiune de operații executate una după alta în ordinea naturală lor, în scopul de a produce date de lucru în date de timp. Operațiile se pot realiza în cadrul aceluși structuri sunt:

- declanșarea datelor cu var f. utilizată
- citirea, prelucrare datelor de intrare
- scrierea datelor de ieșire a rezultatelor
- operații de selecție în ceea ce privește variabile (se poate atribui o valoare constantă, o expresie (ordoni matematici) sau o altă variabilă
- o combinație liberă a operațiilor precedente care să se execute consecutiv.

Schemă logică

```

graph TD
  I1 --> I2
  I2 --> I3
  
```

Paradigmă

var f
 n variabile, o expresie
 I1, I2, ... în ordinea de execuție
 I1
 I2
 I3

4

Operații de selecție

Schemă logică	Simbol pseudocod
	$\text{if } \text{var} = \text{val} \text{ then } \text{var} = \text{val} \text{ else } \dots \text{ else } \text{val} \text{ fi}$
Operații de atribuire	
	$\text{if } \text{var} = \text{val} \text{ then } \text{var} = \text{val} \text{ else } \text{var} = \text{val} \text{ fi}$
Operații de atribuire	
	$\text{var} = \text{val}$

Exemple: calculul sumei, produsului și a mediei aritmetice a trei numere reale.

2. Structura decizională

2.1. Structura decizională simplă

Structura alternativă simplă. În acest caz se va executa doar unul din cele două blocuri de instrucțiuni în funcție de valoarea de adevăr a unei expresii logice.

Exemple

- DACĂ am prezentat la toate materiile **ATENȚIE** nu voi fi în tur
- ALTEL** caz va fi tur
- vârsta mea este mai mică: $a < 18$
- vârsta mea este mai mare: $a > 18$
- vârsta mea este egală: $a = 18$

2.1.1. Structura decizională completă

Structura alternativă decizională completă permite alegerea unei opțiuni sau a unei de două alternative posibile. Prin condiția logică și propoziția care să poată evalua ca adevărată sau falsă. Condiția pot fi **expresii** prin operații logice (și, sau) și **expresii** prin **no**.

5

Schemă logică

```

graph TD
  C{c} -- SI --> I2
  C -- DA --> I1
  I2 --> J(( ))
  I1 --> J
  
```

Paradigmă

Dacă c = valori {I1}
 altfel {I2}
 o condiție
 I1, I2 sunt de instrucțiuni
 IF c = val {I1}
 else {I2}

Principii de concepție: Se evaluează condiția c; dacă aceasta este adevărată se execută unul din instrucțiunile de pe ramura DA (I1) și se ține din structura; dacă aceasta este falsă se execută unul din instrucțiunile de pe ramura SI (I2) și se ține din structura.

Exemple: Se citește de la tastatură două numere întregi a și b. Dacă ambele numere sunt pozitive și se calculează suma lor altfel se calculează produsul lor.

2.1.2. Structura decizională cu ramură vidă

Structura alternativă cu o ramură vidă (structura decizională cu o ramură vidă) presupune evaluarea unei expresii logice și executarea unei opțiuni din structura; dacă în cazul în care este îndeplinită o anumită condiție, în caz contrar nu se va face nimic activ. Prin condiția logică și propoziția care să poată evalua ca adevărată sau falsă. Condiția pot fi **expresii** prin operații logice (și, sau) și **expresii** prin **no**.

Variantă B

Schemă logică

```

graph TD
  C{c} -- SI --> I1
  C -- DA --> J(( ))
  I1 --> J
  
```

Paradigmă

Dacă c = val {I1}
 o condiție
 I1 și I2 instrucțiuni
 IF c = val {I1}

Principii de concepție: Se evaluează condiția c; dacă aceasta este adevărată se ține din structura; dacă aceasta este falsă se execută unul din instrucțiunile I și se ține din structura.

Exemple: Se citește de la tastatură două numere întregi a și b. Dacă ambele numere cu sunt pozitive și se calculează produsul lor.

6

Mai multe detalii se găsesc în [pagina documentului din Biblioteca.RegieLive.ro](http://pagina.documentului.din.Biblioteca.RegieLive.ro)