

3. Algoritmi

3.1. Reprezentarea algoritmilor

3.1. Reprezentarea algoritmilor

- Un algoritm este un concept abstract. Reprezentarea unui algoritm înseamnă implementarea lui fizică.
- Un algoritm se poate reprezenta fizic în mai multe moduri. Exemplu: algoritmul de rezolvare a ecuației de gradul 1 se poate reprezenta prin:
 - Calculele efectuate pe hârtie
 - Prin circuite electronice dacă s-ar construi o mașină care să rezolve această ecuație
 - Printr-un program care descrie pentru un calculator în instrucțiunile lui, algoritmul de rezolvare a ecuației

Reguli de elaborare a unui algoritm

- Să definiți datele de intrare, datele de ieșire și datele intermediare
- Să definiți operațiile care se vor executa cu datele asupra cărora acționează algoritmul
- Să definiți noțiunea de structură de control a algoritmului
- Să definiți succesiunea de structuri care formează algoritmul.

- Modul în care se descrie un algoritm ce rezolvă o problemă cu ajutorul calculatorului trebuie să fie clar deoarece calculatorul nu poate opta singur pentru o anumită posibilitate.
- Un algoritm se poate descrie în **pseudocod** (un limbaj apropiat de un limbaj de programare dar neformalizat și neconstrans de reguli de sintaxă).
- Pseudocodul nu poate fi folosit ca limbaj de programare pentru că nu poate exprima instrucțiunile care se dau calculatorului pentru a rezolva problema descrisă de algoritm.

- **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**.

- Pseudocodul permite și descrierea datelor asupra cărora acționează algoritmul. Pentru precizarea tipului de dată se folosesc cuvinte cheie. De exemplu, se pot folosi următoarele cuvinte cheie:
 - **întreg** - tipul numeric întreg;
 - **real** - tipul numeric real;
 - **logic** - tipul logic;
 - **caracter** - tipul caracter;
 - **șir** - tipul șir caracter.
- Cuvântul cheie care precizează tipul este urmat de o listă prin care se enumeră identificatorii datelor care corespund aceluși tip.

- 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 (un mesaj reprezentat printr-o constantă de tip șir de caractere). Elementele listei se separa prin virgulă
- **Exemplu:** **citește** a,b;
scrie "soluția ecuației este ",a/b;

- Cuvintele cheie pot fi în română sau în engleză. Varianta în engleză este mai apropiată de cuvintele cheie folosite în limbajul de programare.
- Pentru a delimita secvența de descriere a datelor de secvența de descriere a pașilor algoritmului, aceștia vor fi încadrați între cuvintele cheie **început** și **sfârșit (begin si end)**.

Exemplu: algoritmul de rezolvare a ecuației de gradul 1. $ax+b=0$

real a,b,x;

început

citește a,b;

dacă a=0

atunci dacă b=0

atunci scrie "ecuația are o infinitate de soluții";

altfel scrie "ecuația nu are soluții";

sfârșit_dacă;

altfel x=-b/a;

scrie "soluția ecuației este ",x;

sfârșit_dacă;

sfârșit.

Observații

- În pseudocod operația de comparație a fost exprimată prin cuvintele cheie **dacă** și **sfârșit_dacă**. Aceste două cuvinte formează o **structură de control**.
- O structură de control este o entitate în cadrul algoritmului prin care se descrie modul în care pașii algoritmului își predau controlul unul altuia.
- În pseudocod, pentru orice structură de control, se folosește o pereche de cuvinte cheie. Primul cuvânt precizează începutul structurii (**dacă**) iar al doilea cuvânt precizează sfârșitul structurii (**sfârșit_dacă**)

- Numărul de structuri deschise trebuie să fie egal cu numărul de structuri închise (numărul de **dacă** trebuie să fie egal cu numărul de **sfârșit_dacă**).
- Pentru e urmări mai ușor structurile este recomandat să scrieți indentat corpul structurii față de cuvintele cheie cu care începeți și terminați structura. Scrieți la același nivel de indentare propozițiile care se execută secvențial (în ordinea în care au fost scrise).

3.2. Principiile programării structurate

- Algoritmul este format din pașii care urmează să se execute și ordinea în care se vor executa pentru a rezolva problema
- Algoritmul pentru rezolvarea ecuației de gradul întâi, conține nouă pași, iar ordinea de executare depinde de valoarea celor doi coeficienți a și b .

- Dacă $a=0$ și $b=0$, se execută în ordine pașii: 1,2,3,4,5,9
- Dacă $a=0$ și $b \neq 0$, se execută în ordine pașii: 1,2,3,4,6,9
- Dacă $a \neq 0$, se execută în ordine pașii: 1,2,3,7,8,9

Pasul 1	Început
Pasul 2	Comunică valorile pentru a și b
Pasul 3	Compara $a=0$. Dacă este adevărat execută pasul 4; altfel, execută pasul 7
Pasul 4	Compara $b=0$; dacă este adevărat execută pasul 5; altfel execută pasul 6
Pasul 5	Comunică mesajul "ecuația are o infinitate de soluții"; mergi la pasul 9
Pasul 6	Comunică mesajul "ecuația nu are soluții"; mergi la pasul 9
Pasul 7	Calculează $x=-b/a$;
Pasul 8	Comunică valoarea lui x ;
Pasul 9	Terminat

Structura de control

- Structura de control a algoritmului definește ordinea de executare a pașilor adică ordinea în care un pas predă controlul altui pas și prin care se determină fluxul controlului.
- În cadrul algoritmilor pot fi folosite trei tipuri de structuri de control:
 - Structura liniară
 - Structura alternativă
 - Structura repetitivă