

## STRUCTURA SECVENȚIALA

Etaplele (pașii) unui algoritm sunt formate din **instrucțiuni**. Ordinea efectuării instrucțiunilor este importantă.

Un șir de instrucțiuni (numit în informatică secvență de instrucțiuni) care se execută fiecare o singură dată, de la prima până la ultima instrucțiune din șir, poartă denumirea de **structură secvențială (liniară)**.

Structura liniara reprezinta un grup de operații care se executa în ordinea scrierii lor:

Daca **S1, S2, ..., Sn** sunt instrucțiuni, atunci

**S1;**

**S2;**

...

**Sn.**

este o structura secvențială (liniara).

Structura secvențială (liniara) presupune executarea instrucțiunilor succesiv, în ordinea în care acestea sunt scrise.

Conform teoremei de structură, orice algoritm poate fi descris prin compunerea a trei structuri fundamentale:

- *structura liniară* (secvențială);
- *structura alternativă* (selecția);
- *structura repetitivă* (repetiția sau ciclul).

### 1. Reprezentarea structurii secvențiale (liniare)

Reprezentarea structurii secvențiale (liniare) se realizează folosind un ansamblu de convenții respectate sistematic, care definesc operațiile permise, numite instrucțiuni, pentru reprezentarea algoritmilor. Se poate utiliza pseudocodul, un limbaj creat mai degrabă pentru a fi înțeles și interpretat de către oameni decât de către calculatoare și schemele logice.

Un program pseudocod poate fi cu ușurință tradus în limbaj de programare de către un informatician.

Structura secvențială impune o anumită **ordine a operațiilor/ instrucțiunilor**. Se începe cu **declararea variabilelor** de care avem nevoie pe parcursul algoritmului, apoi se **citesc datele** cunoscute ale problemei, urmează **efectuarea calculelor** în funcție de cerințe și la final **scrierea rezultatelor** obținute.

Instrucțiunile utilizate sunt:

- **Declararea datelor**

**Tip variabila;**

La începutul oricărui algoritm, vom preciza datele de intrare, datele de ieșire, eventualele date de manevră, precum și tipul acestora. Înainte de a utiliza orice variabilă, o vom declara, precizând numele și tipul ei. O variabilă nu poate fi declarată de mai multe ori în același algoritm.

*Exemple:*

x real;

c caracter;

i întreg;

sau real x; caracter c; întreg i;

- **Operația de citire**

**Citește** variabila<sub>1</sub>, variabila<sub>2</sub>, ..., variabila<sub>n</sub>;

*Efect:* Prin operația de citire (denumită și operație de intrare) se preiau succesiv valori de la tastatură și se asociază, în ordine, variabilelor specificate.

- **Operația de atribuire**

variabila  $\leftarrow$  expresie;

*Efect:* se evaluează expresia, apoi se atribuie valoarea expresiei variabilei din membrul stâng.

- **Operația de scriere**

**Scrie** expresie<sub>1</sub>, expresie<sub>2</sub>, ..., expresie<sub>n</sub>;

*Efect:* Operația de scriere (denumită și operație de ieșire) presupune evaluarea în ordine a expresiilor specificate și afișarea pe ecran a valorilor.

Parcurgerea instrucțiunilor în secvență, în ordinea specificării lor, reprezintă o **structură liniară** (secvențială).

## 2. Structura secvențială (liniară). Reprezentare cu scheme logice

Corespunzător instrucțiunilor structurii secvențiale descrise în pseudocod (citire, scriere, atribuire), acestea au echivalent în blocurile unei scheme logice.

- Blocurile utilizate în schema logică sunt:
- Bloc START/ STOP
- Bloc de citire (primire a datelor de intrare)
- Bloc de scriere (furnizare a datelor de ieșire)
- Bloc de calcul - atribuire (prin care o variabilă primește ca valoare rezultatul obținut prin evaluarea unei expresii, valoarea altei variabile sau valoarea unei constante)
- Săgeți de legătură



## 3. Exemplu de algoritm cu structură secvențială descris în limbaj natural, pseudocod și schemă logică

*Calculează perimetrul și aria unui dreptunghi, dacă știi lungimea și lățimea.*

### 3.1. limbaj natural:

**Pasul 1:** În enunț se cere să calculezi perimetrul și aria unui dreptunghi.

**Pasul 2:** Pentru a calcula perimetrul și aria unui dreptunghi trebuie să cunoști mărimea celor două laturi: lungimea și lățimea. Așadar, datele de intrare sunt mărimile celor două laturi, iar datele de ieșire reprezintă perimetrul și aria dreptunghiului.

**Pasul 3:** Dacă lățimea este de 3 cm și lungimea de 4 cm, atunci perimetrul este suma laturilor și aria este produsul lor: 14 cm și respectiv 12 cm<sup>2</sup>.

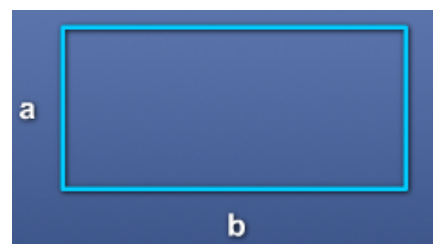
**Pasul A:** Citești prima latură a;

**Pasul B:** Citești a doua latură b;

**Pasul C:** Calculezi perimetrul ca fiind egal cu  $a+b+a+b$ ;

**Pasul C:** Calculezi aria ca fiind egală cu  $a*b$ ;

**Pasul D:** Afișezi perimetrul și aria;



### 3.2. Pseudocod:

```
intreg a,b,p,s;  
citește a;  
citește b;  
 $p \leftarrow 2*a+2*b$ ;  
 $s \leftarrow a*b$ ;  
scrie "p= ", p, " s= ", s;
```

### 3.3. Schema logică

