

Structuri de bază. Structura liniară și structura alternativă

Structuri de bază. Structura liniară și structura alternativă

- ▶ Introducere
- ▶ Structuri de bază
 - Structura liniară
 - Structura alternativă
- ▶ Implementarea structurilor de bază
- ▶ Probleme propuse
- ▶ Bibliografie

▶ Introducere

Principiile programării structurate

Prin structură se înțelege o anumită formă de îmbinare a operațiilor cu care lucrează algoritmi.

Programarea structurată are la bază o justificare matematică, furnizată de Böhm și Jacopini și cunoscută ca **teorema de structură** care precizează că orice algoritm având o *intrare* și o *ieșire* (adică un singur punct de început și un singur punct de terminare a execuției) poate fi reprezentat ca o combinație a trei structuri de control:

- a. **secvența** – succesiune de două sau mai multe operații (**structura secvențială** sau **structura liniară**);
- b. **decizia** – alegerea unei operații din două alternative posibile (**structura decizională** sau **structura alternativă**);
- c. **repetiția** – repetarea unei operații atâta timp cât o anumită condiție este îndeplinită – ciclul cu test inițial (**structura repetitivă** sau **structura iterativă**).

În programarea structurată se mai admit trei tipuri de structuri de control:

- selecția;
- ciclul cu test final;
- ciclul cu contor.

Operații specifice algoritmilor

- operațiile de bază care pot fi efectuate de un algoritm:
 - atribuire
 - operația de citire (intrare)
 - operația de scriere (ieșire)

Atribuire

variabilă ← expresie

Structuri de bază. Structura liniară și structura alternativă

Exemple:

suma $\leftarrow a + b$

media $\leftarrow (x+y+z) / 3$

Operația de citire

citește variabila₁, variabila₂, ... , variabila_n

Exemple:

citește a, b

citește x, y, z

Operația de scriere

scrie expresie₁, expresie₂, ... , expresie_n

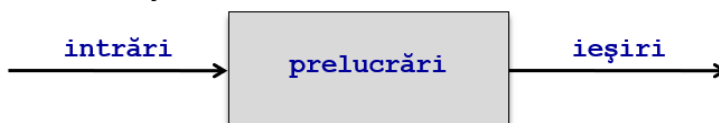
Exemple:

scrie suma

scrie suma, ' ', media

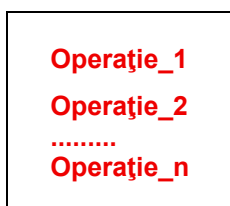
► Structuri de bază

Structurile de bază sau **structurile de control** sunt secvențe de instrucțiuni care controlează fluxul de execuție al unui algoritm, ajutând la structurarea logică a acestuia.



1. Structura liniară (structura secvențială)

- operațiile se execută în ordine una după alta de la prima operație până la ultima operație
- secvența reprezintă o succesiune de două sau mai multe operații care conține o transformare de date
- execuția se descrie astfel:



Structuri de bază. Structura liniară și structura alternativă

Principiul de execuție:

- se execute operație_1
- se execute operație_2
- ...
- se execute operație_n

Exemplu:

Să se calculeze și să se afișeze pe ecran suma, produsul și diferența a trei nume întregi x , y și z citite de la tastatură.

Date de intrare: x , y , z

Date de ieșire: S , P , D

citește x, y, z (numere întregi)

$S \leftarrow x + y + z$

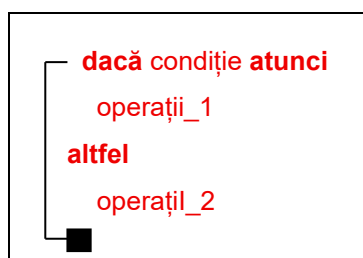
$P \leftarrow x * y * z$

$D \leftarrow x - y - z$

scrie $S, ' ', P, ' ', D$

2. Structura alternativă (structura decizională)

- decizia permite alegerea unei operații/secvențe de operații din două alternative posibile
- execuția se descrie astfel:



Principiul de execuție:

Pasul 1: se evaluează condiția

Pasul 2: - dacă condiție este adevărată se va executa operații_1

- dacă condiție este falsă se execute operații_2

Observații:

- se va executa operații_1 sau operații_2

Structuri de bază. Structura liniară și structura alternativă

► Implementarea structurilor de bază

Structura alternativă se implementează în limbajul C++ folosind instrucțiunea **if**.

Instrucțiunea **if**

Forma instrucțiunii:

```
if(expresie)
{
    operații_1;
}
else
{
    Operații_2;
}
```

Principiul de execuție:

Pasul 1: se evaluează condiția

Pasul 2: - dacă condiție este adevărată se va executa `operații_1`

- dacă condiție este falsă se execute `operații_2`

Exemplu:

Se dau două numere naturale **a** și **b** distincte. Să se determine care dintre ele are valoarea mai mare.

Date de intrare: **a, b**

Date de ieșire: **maxim**

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,maxim;
    cout<<"Introdu a:";
    cin>>a;
    cout<<"Introdu b:";
    cin>>b;
    if(a>b)
    {
```

Structuri de bază. Structura liniară și structura alternativă

```
        maxim=a;
    }
    else
    {
        maxim=b;
    }
    cout<<"Maximul este:"<<maxim<<endl;
    return 0;
}
```

Instrucțiunea if cu varianta unei căi nule

Forma instrucțiunii:

```
if(expresie)
{
    operații;
}
```

Principiul de execuție:

Pasul 1: se evaluează condiția

Pasul 2: - dacă condiție este adevărată se va executa operații

- dacă condiție este falsă execuția se încheie

Exemplu:

Se dau două numere naturale a și b . Să se determine câtul împărțirii lui a la b .

Date de intrare: a, b

Date de ieșire: rezultatul $a : b$ sau fără rezultat dacă $b=0$

```
#include <iostream>

using namespace std;

int main()
{
    int a,b;
    cout<<"Introdu a:";
    cin>>a;
```

Structuri de bază. Structura liniară și structura alternativă

```
    cout<<"Introdu b:";  
    cin>>b;  
    if (b!=0)  
    {  
        cout<<a/b;  
    }  
    return 0;  
}
```

► **Probleme propuse**

Pbinfo:

✎ Probleme:

- ✓ structura liniară:
#939, #1260, #175, #2201, #102, #169, #2601, #2264, #467, #209
- ✓ structura alternativă:
#4861, #4862, #109, #469, #106, #4339, #9, #452, #1311, #3275

► **Bibliografie**

1. Cerchez E., Șerban M., *Programarea în limbajul C/C++ pentru liceu*, Editura Polirom, Iași, 2005
2. <https://www.pbinfo.ro/>
3. <https://www.ududec.com/>