

Programarea Structurată

Este programarea ce folosește structuri fundamentale ce reprezintă primul nivel de abstractizare în programare.

Etape de rezolvare a problemelor cu calculatorul

- Rezolvarea problemelor cu calculatorul se face prin conceperea unui program de calculator în etape:
- 1. **Analiza problemei**= stabilirea datelor de intrare și de ieșire.
- 2. **Raționamentul problemei**= descoperirea relațiilor dintre datele de intrare și datele de ieșire.
- 3. **Reprezentarea algoritmului**= scrierea lui în pseudocod.
- 4. **Realizarea programului**= Programul de calculator reprezintă traducerea unui algoritmi într-un limbaj de programare.
- 5. **Compilarea programului**= pot apărea erori de sintaxă.
- 6. **Rularea programului**= pot apărea erori logice.
- Algoritmi sunt formați din structuri ce devin instrucțiuni în limbajul de programare ales.
- Datorită folosirii structurilor din algoritmi programarea se numește programare structurată.

Algoritm – structuri C++ - instrucțiuni

- **-liniară**
- **secvențiale**
- se execută o singură dată necondiționat
- cin>>
- -citire
- cout<<
- -scriere
- a=b+c; "=" operator binar
- -atribuire

Algoritm – structuri C++ - instrucțiuni

- **-alternativă**
- **if(condiție)**
{...instrucțiuni secvențiale...}
- se alege continuarea programului în funcție de evaluarea unei condiții
- else
{...instrucțiuni secvențiale...}

Algoritm – structuri C++ - instrucțiuni

- Repetitivă cu test inițial **RTI**
- Repetitivă cu test final **RTF**
- cât timp (condiție) execută
- while(condiție) {... instrucțiuni secvențiale...}
- Cât timp este adevărată o condiție se execută un bloc de instrucțiuni

Algoritm – structuri C++ - instrucțiuni

- Repetitivă cu test inițial **RTI**
- Repetitivă cu test final **RTF**
- cât timp (condiție) execută
- do {... instrucțiuni secvențiale...} while(condiție);
- Se execută un bloc de instrucțiuni cât timp este adevărată o condiție

Algoritm - structura RC

- Repetitivă cu contor **RC** pentru(initializare;finalizare;incrementare/decrementare) execută

Plecând de la o valoare inițială (initializare) până la o valoare finală (finalizare) se incrementează sau se decrementează un contor.

C++ - instrucțiunea for

```
for(=0;j<=n;i++)
{... instrucțiuni secvențiale...}
```

i++ echivalent i=i+1

i-- echivalent i=i-1

I este contorul iar n este valoarea finală

Legea lui Bohm-Jacoppini

- Orice program oricât de complex se poate scrie folosind doar 3 structuri:
 - ✓ Liniară
 - ✓ Alternativă
 - ✓ Repetitivă cu test inițial RTI

Consecință:

Orice structură repetitivă (RTF sau RC) se poate rescrie ca structură cu test inițial.