



Structura alternativa

- Structura alternativă apare în practică în două variante și anume:

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

•**Structura alternativă multiplă:** în această situație se va executa un bloc de instrucțiuni din mai multe variante posibile.

Structura alternativă

Auzim în viața de zi cu zi afirmații de genul:

DACĂ am promovat la toate materiile

ATUNCI

ma voi duce în tabără,

ALTFEL

stau să învăț.

SFÂRȘIT DACĂ

Se remarcă aici trei cuvinte ce au un rol deosebit: DACĂ, ATUNCI, ALTFEL.



- o **conditie**, transcrisă prin “**am promovat la toate materiile**”, conditie pe care o vom nota cu **C**;
- o **acțiune** transcrisă prin mesajul “**mă voi duce în tabără**”, notata cu **B1**, acțiune asociata cu cuvântul **ATUNCI**, adică se execută dacă și numai dacă “**am promovat la toate materiile**”;
- o **acțiune** transcrisă prin mesajul “**stau să învăț**”, notată cu **B2**, acțiune asociată cu cuvântul **ALTFEL**, adică se execută dacă și numai dacă **NU** “**am promovat la toate materiile**”

Structura alternativă este implementată în limbajul C++ de instrucțiunea **IF**. În continuare vom prezenta modul de reprezentare a acestei structuri în limbaj pseudocod, în schemă logică și în limbajul de programare C++. Modul de execuție este identic pentru oricare din cele trei reprezentări. Sintetizând cu notațiile de mai sus structura alternativă se poate reprezenta astfel:



Schema logică	Pseudocod	Instructiune C++
<pre> graph TD Start(()) --> C{C} C -- Nu --> B2[B2] C -- Da --> B1[B1] B2 --> Join(()) B1 --> Join Join --> Exit(()) </pre>	<pre> dacă c atunci B1 altfel B2 </pre>	<pre> if(expresie logica) instructiune1; else instructiune2; </pre>

Pot exista și cazuri particulare de utilizare a acestei structuri, atunci când unul dintre cele două blocuri de instrucțiuni lipsește.

Cazul 1. Dacă lipsește B2 atunci structura se reprezintă astfel:

Schema logică	Pseudocod	Instructiune C++
<pre> graph TD Start(()) --> C{C} C -- Da --> B1[B1] C -- Nu --> Join(()) B1 --> Join Join --> Exit(()) </pre>	<pre> dacă c atunci B1 </pre>	<pre> if(expresie logica) instructiune1; </pre>



Cazul 2. Daca lipseste B1 atunci structura se reprezinta astfel:

Schema logică	Pseudocod	Instructiune C++
<pre> graph TD Start(()) --> C{C} C -- Da --> B1[B1] C -- Nu --> Merge(()) B1 --> Merge Merge --> End(()) </pre>	dacă C atunci B1	<pre>if(expresie logica) instructiune1;</pre>

Pot exista și situații în care structurile alternative sunt incluse unele în altele. În acest caz spunem că instrucțiunile if sunt **imbricate**. Există o regulă de asociere a cuvântului else de if:

„În situația instrucțiunilor if imbricate, și atunci când pentru gruparea instrucțiunilor nu se folosesc acolade cuvântul else se asociază primului cuvânt if aflat la stanga lui. (asemanator cu asocierea închiderii corecte a grupurilor de paranteze din matematica).”



În exemplul următor este sugerat cum sunt imbricate 3 instructiuni if. Pentru a le putea identifica ușor ele sunt scrise cu culori diferite (negru, albastru si portocaliu).

```
if(c1)
  if(c2)
    if(c3)
      instructiunea1;
    else
      instructiunea2;
  else
    instructiunea 3
else
  instructiunea 4;
```

se execută când c2 adevărată

se execută când c1 adevărată



Dacă în aceasta schemă grupăm instrucțiunile cu acolade se poate schimba fundamental structura programului:

```
if(c1)
{
    if(c2)
    {
        if(c3)
        {
            instructiunea1;
        }
        else
        {
            instructiunea2;
        }
    }
    else
    {
        instructiunea 3
    }
}
```

se execută când c2 adevarata

se execută când c1 adevarata
c2 falsa

Se observă că primul cuvânt **else** este asociat cu cel de-al doilea **if**, acest lucru fiind determinat de gruparea între acolade a instrucțiunii **if (c3)**, reprezentate pe desen cu portocaliu