

Structura repetitivă

Instrucțiunea repetitivă condiționată anterior

Cât-timp expresie **execută**
instrucțiune;

Pas 1: se evaluează expresia;

Pas 2: dacă valoarea expresiei este *fals*, se iese din instrucțiunea *cât-timp*; dacă valoarea expresiei este *adevărat*, se execută instrucțiunea, apoi se revine la Pas 1.

Observații:

1. Instrucțiunea se execută repetat, cât timp valoarea expresiei este *adevărat*. Pentru ca ciclul să nu fie infinit, este obligatoriu ca instrucțiunea care se execută să modifice cel puțin una din variabilele care intervin în expresie, astfel încât aceasta să poată lua valoarea *fals*.
2. Dacă expresia are de la început valoarea *fals*, instrucțiunea nu se repetă nici măcar o dată.

Exemplu: Să se realizeze un algoritm ce calculează cel mai mare divizor comun a două numere.

Date de intrare: a, b naturale;

Date de manevră: rest natural;

Date de ieșire: b natural;

Citește a, b;

Cât-timp $b \neq 0$ **execută**

```
{  
    rest  $\leftarrow$  a%b;  
    a  $\leftarrow$  b;  
    b  $\leftarrow$  rest;  
}
```

Scrie "C.m.m.d.c. este ", a;

Instrucțiunea repetitivă condiționată posterior

Execută

instrucțiune

cât-timp expresie;

Pas 1: se execută instrucțiunea;

Pas 2: se evaluează expresia;

Pas 3: dacă valoarea expresiei este *fals*, se iese din instrucțiunea *cât-timp*; dacă valoarea expresiei este *adevărat*, se revine la Pas 1.

Observații:

1. Instrucțiunea se execută repetat, cât timp valoarea expresiei este *adevărat*. Pentru ca ciclul să nu fie infinit, este obligatoriu ca instrucțiunea care se execută să modifice cel puțin una din variabilele care intervin în expresie, astfel încât aceasta să poată lua valoarea *fals*. Deoarece evaluarea expresiei se face după execuția instrucțiunii, instrucțiunea se execută cel puțin o dată.
2. Instrucțiunea repetitivă condiționată posterior (*Execută cât-timp*) poate fi simulată cu ajutorul instrucțiunii repetitive condiționate anterior (*Cât-timp*) și reciproc.

Exemplu: Să se realizeze un algoritm ce calculează cel mai mare divizor comun a două numere.

Date de intrare: a, b naturale;

Date de manevră: rest natural;

Date de ieșire: b natural;

Citește a, b;

Execută

{

rest $\leftarrow$ a%b;

a $\leftarrow$ b;

b $\leftarrow$ rest;

}

Cât-timp rest $\neq$ 0;

Scrie "C.m.m.d.c. este ", a;

- Simularea instrucțiunii *Cât-timp* cu ajutorul instrucțiunii *Execută cât-timp*:

```
Dacă expresie atunci  
    Execută  
        instrucțiune  
    Cât-timp expresie;
```

Observați că a fost necesară testarea inițială a expresiei deoarece, spre deosebire de instrucțiunea *Cât-timp*, instrucțiunea *Execută cât-timp* efectuează o dată *instrucțiune* înainte de a testa expresia.

- Simularea instrucțiunii *Execută cât-timp* cu ajutorul instrucțiunii *Cât-timp*:

```
    instrucțiune;  
    Cât-timp expresie execută  
        instrucțiune;
```

Observați că în acest caz a fost necesar să executăm o dată instrucțiunea în afara ciclului.

Deducem că nu este necesară existența ambelor instrucțiuni, dar în funcție de problemă, vom alege instrucțiunea repetitivă adecvată, pentru care descrierea algoritmului este mai concisă.

Instrucțiunea repetitivă cu număr finit de pași

Pentru $\text{contor} \leftarrow \text{expresie_1}$, expresie_2 **execută**
instrucțiune

Pas 1: se evaluează *expresie_1*.

Pas 2: se atribuie variabilei *contor* valoarea expresiei *expresie_1*.

Pas 3: se evaluează *expresie_2*.

Pas 4: dacă valoarea variabilei *contor* este mai mare decât valoarea expresiei *expresie_2*, atunci se iese din instrucțiunea repetitivă. Dacă valoarea variabilei *contor* este mai mică sau egală cu valoarea expresiei *expresie_2*, atunci se execută *instrucțiune* și apoi se incrementează (se mărește cu 1) valoarea variabilei *contor*, după care se revine la Pas 3.

Observații:

1. Dacă inițial valoarea expresiei *expresie_1* este mai mare decât valoarea expresiei *expresie_2*, *instrucțiune* nu se execută nici o dată. În caz contrar, și dacă *instrucțiune* nu modifică valorile variabilelor care intervin în *expresie_2*, *instrucțiune* se execută de $(\text{expresie_2} - \text{expresie_1} + 1)$ ori.
2. Instrucțiunea repetitivă cu număr cunoscut de pași poate fi simulată cu ajutorul celorlalte două instrucțiuni repetitive.

Exemplu: Să se realizeze un algoritm ce calculează suma primelor n numere naturale.

Date de intrare: n natural;

Date de manevră: i natural;

Date de ieșire: s natural;

Citește n ;

$s \leftarrow 0$;

Pentru $i \leftarrow 1$, n **execută**

$s \leftarrow s + i$;

Scrie „Suma este ”, s ;

- Simularea instrucțiunii *Pentru* cu ajutorul instrucțiunii *Cât-timp*:

```
contor ← expresie_1;  
Cât-timp contor ≤ expresie_2 execută  
{  
    instrucțiune;  
    contor ← contor + 1;  
}
```

- Simularea instrucțiunii *Pentru* cu ajutorul instrucțiunii *Execută cât-timp*:

```
contor ← expresie_1;  
Dacă contor ≤ expresie_2 atunci  
    Execută  
    {  
        instrucțiune;  
        contor ← contor + 1;  
    }  
cât-timp contor ≤ expresie_2;
```

Executarea repetată a unei instrucțiuni, controlată de valoarea unei expresii, reprezintă o **structură repetitivă**.