

**Material didactic pentru studenți la  
disciplina “Bazele programării”**

# **Instrucțiunea While**

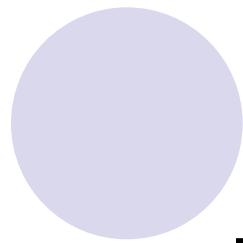


**Lecție elaborată de Bacalîm Alina  
masterandă gr.IP21M**

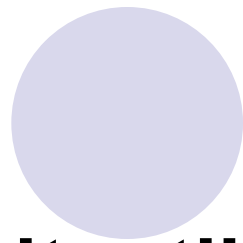
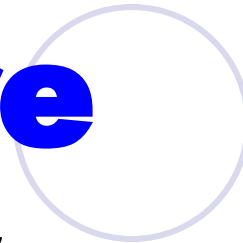
**13.04.2012**

# Cuprins

- Noțiuni generale
- Sintaxa instrucțiunii **Sintaxa**  
**instrucțiunii** Repeat
- Principiu de lucru
- Schema-bloc
- Exemplu rezolvat
- Concluzii
- Temă pentru acasă



# Introducere



Deseori în viață ne confruntăm cu situații care au ca bază întrebarea: **cît timp?**

**De exemplu:** Fie că avem de turnat apă într-un vas.

**Întrebare:** Cît timp vom turna apă?

**Răspuns:** Voi turna apă atîta timp cînd vasul nu va fi plin.

Această acțiune este repetitivă, în care nu cunoaștem numărul de repetări, la fiecare turnare se verifică dacă paharul nu e plin, dacă nu e plin se mai toarnă în el apă, dar dacă e plin acțiunea de turnare ia sfârșit.

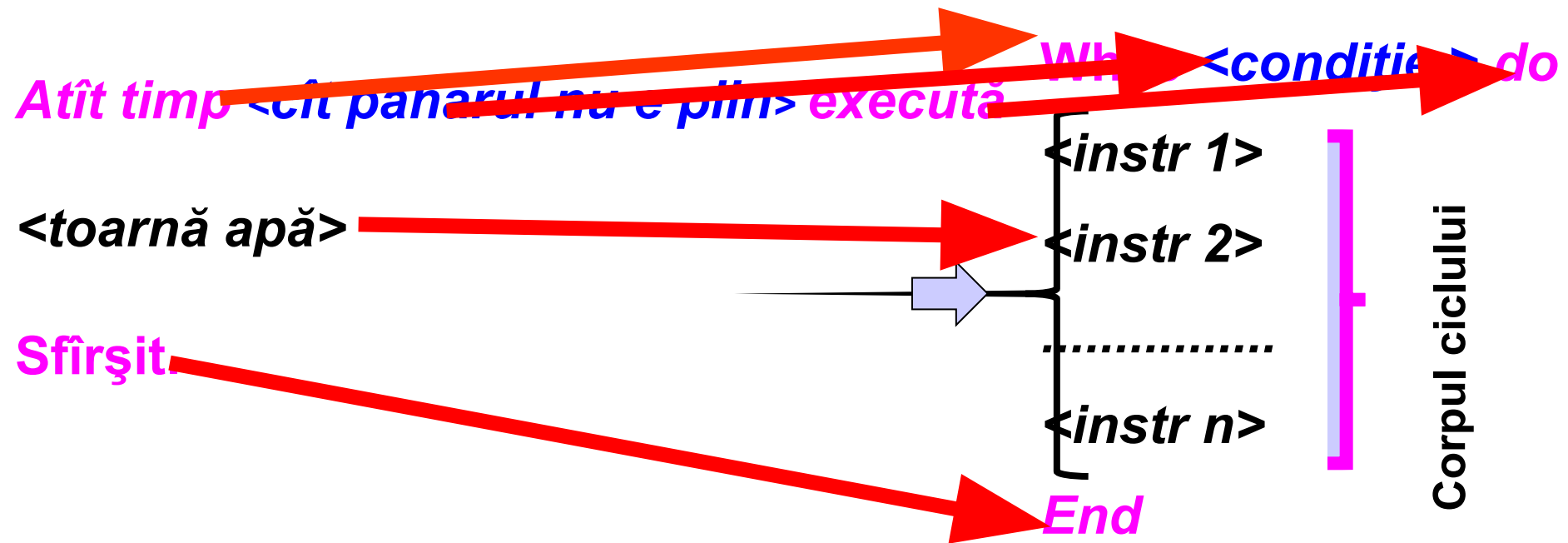


În mod algoritmic astfel de situații se soluționează cu ajutorul **instrucțiunii While**.

**Instrucțiunea While** este o construcție de ciclare cu ***test inițial*** sau cu test condiționat anterior, deoarece verificarea condiției are loc la începutul ciclului.

# Sintaxa instructiunei While

Transformăm exemplu analizat în mod  
algoritmico:





**<conditie>** este o expresie logica care reprezintă condiția de ieșire din ciclu.

**<instruct1>, <instruct2>, ..., <instruct n>** reprezintă secvențe de instrucțiuni, care și reprezintă corpul ciclului

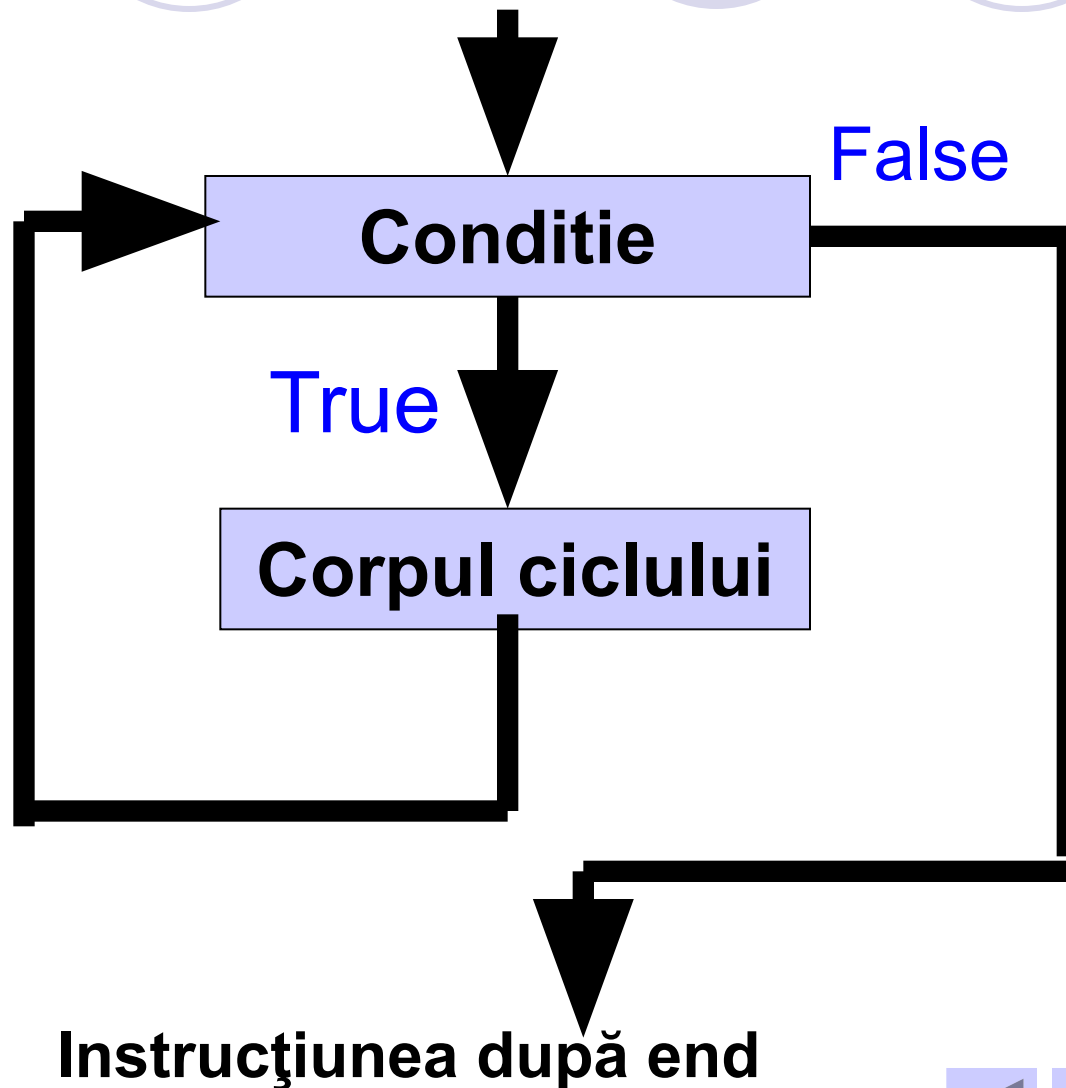
***Corpul ciclului*** se execută atîta timp cît condiția este adevărată.

# Principiu de lucru


1. se evaluează condiția;
2. dacă condiția este **adevărată (true)**, atunci se execută corpul ciclului, apoi se trece la punctul 1;
3. dacă condiția este **falsă**, atunci are loc ieșirea din ciclu și gestiunea se transmite la următoarea instrucțiune după **end**.



# Schema-bloc



În concluzie, corpul ciclului se va executa atîta timp cît valoarea condiției de ciclare este **adeverată**, ieșirea din ciclu avînd loc cînd expresia devine **falsă**. Dacă, la intrare în ciclu, condiția de ciclare nu este îndeplinită (expresia are valoarea False), corpul ciclului nu se va executa niciodată.



**Dacă se știe de câte ori se repetă corpul ciclului, atunci așa tip de ciclu se numește **ciclul cu contor**. El se termină atunci când toate iterațiile au fost executate.**

**Pentru numărarea iterațiilor în ciclul cu contor se folosește o variabilă, numită **variabila ciclului**. Valoarea acestei variabile se incrementează la fiecare iterație, iar în condiția ciclului se compară cu numărul dat de repetări.**

# Schema generală de organizare a ciclului cu contor realizat cu While

```
i:= 0;  
While i< N do  
begin  
<corpul ciclului>  
i:= i+1;  
End.
```

# Exemplu

Să se calculeze suma a 5  
numere întregi citite de la  
tastatură.

*Program Suma;*

*Var S: integer;  
Num: integer;  
i: integer;*

*Begin*

*S:=0;*

*i:=1;*

*While i < 5*

*Write ('Dati numarul Num=');*

*Readln(Num);*

*S:=S+Num;*

*i:=i+1;*

*end;*


*Writeln ('Suma este ', S);*

*Readln;*

*End.*

Suma este 15

Num	S	i	i=5
	0	0	
2 →	2	1	1=5 -> false
↓ 3 →	5	2	2=5 -> false
↓ 1 →	6	3	3=5 -> false
↓ 4 →	10	4	4=5 -> false
↓ 5 →	15	5	5=5 -> true



**Destul de frecvent în programare, un set de instrucțiuni se repetă nu de un număr dat de ori, ci pînă cînd se îndeplinește o anumită condiție.**

**De exemplu: de la tastatură se citesc numere întregi pînă la apariția primului număr negativ. Dacă numărul se va citi în variabila Num, atunci ciclul se va repeta atîta timp cît  $\text{Num} \geq 0$ .**

# Schema generală de organizare a ciclului cu un număr necunoscut de pași realizat cu While

**Write ('x=');**

**Readln (x);**

**While not** <conditia de terminare> **do**

While (Not <condiția de t

**begin**

<p

**<prelucrare x>**

**Readln (x);**

**End.**

# Exemplu

De la tastatura se introduc numere intregi pina la aparitia primului numar negativ. Să se calculeze suma lor.

```
Program P2;  
Var S: integer;  
Num: integer;  
Begin  
S:=0;  
Write ('Num=');  
Readln (Num);  
While Num >= 0 do  
begin  
S:= S + Num;  
Write ('Num=');  
Readln (Num);  
End;  
Writeln ('Suma = ', S);  
Readln;  
End.
```

Num	Num >= 0	S
1	1 >= 0 F	0
2	2 >= 0 F	0
-1	-1 >= 0 T	-1
-3	-3 >= 0 T	- 4
4	4 >= 0 F	- 4
0	0 >= 0 T	- 4
2	-2 >= 0 F	- 6

Se va afisa - 6





***Dacă în algoritm se folosește ciclul While e necesar de determinat :***

- 1. ce se face la o iterație. Setul de instrucțiuni care realizează iterația va forma corpul ciclului;***
- 2. în ce condiții iterațiile se vor executa; condițiile vor forma condiția ciclului;***
- 3. ce trebuie de făcut ca prima iterație să se execute corect. Va fi necesar ca aceste instrucțiuni să fi e programate pînă la intrare în ciclu.***

# Temă pentru acasă

- ◆ Să se elaboreze programul care citește numere întregi până la introducerea numărului 0 și afișează media aritmetică a numerelor introduse.
- ★ \*\*\* Să se elaboreze programul care citește de la tastatură un număr întreg. Să se calculeze suma cifrelor numărului citit.

De exemplu, pentru numărul 324 să obținem  
Suma = 9, adică  $3+2+4=9$ .

● *Vă Doresc*  
*Succes!!!*

