INFORMATICĂ · CLASA A VI-A · STRUCTURI REPETITIVE · LECȚIA 4

Structura repetitivă condiționată posterior (repetă ... până când)

Durată: 45 min Nivel: gimnaziu Înainte: lecțiile g6-03

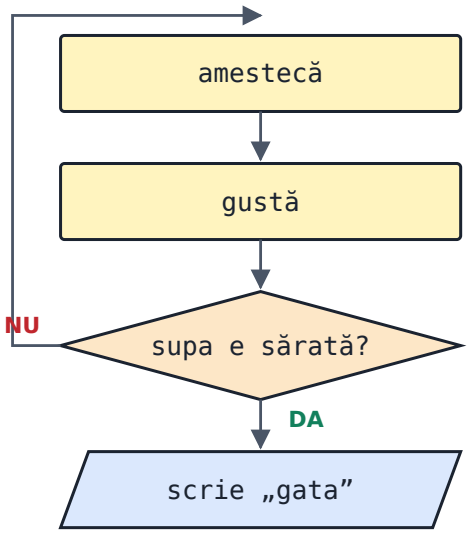
CE VEI ȘTI DUPĂ LECȚIE

- Înțelegi structura în care testul se face DUPĂ pași (pașii se fac cel puțin o dată).
- Deosebești „cât timp” (test la început) de „repetă ... până când” (test la final).
- Simulezi în Scratch o repetiție cu test final.
- Compari cele trei structuri repetitive.

1. Testul la final

„**Repetă**: amestecă supa; gustă; **până când** e destul de sărată.” Mai întâi faci pașii, **apoi** verifici. Pașii se execută **cel puțin o dată**.

```
repetă
    amestecă
    gustă
până când supa e sărată
```



testul se face DUPĂ pași: pașii se execută cel puțin o dată

Schema logică

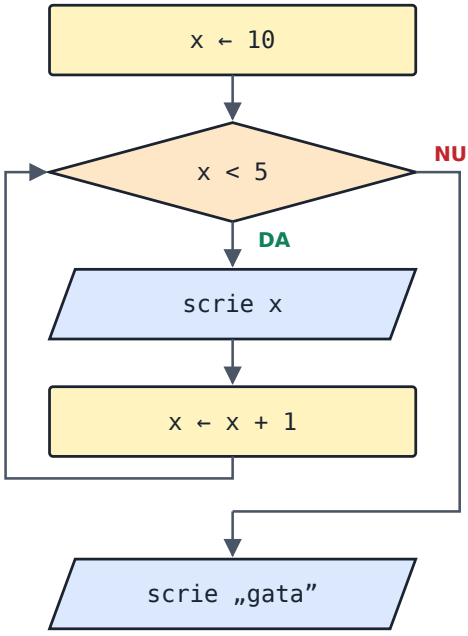
DE CE AȘA?

De ce testul vine după? Uneori nu poți ști răspunsul înainte să încerci: nu poți gusta supa dacă nu ai amestecat-o. La parolă, nu poți verifica parola înainte s-o ceri. Când ai nevoie de informația din interior ca să testezi, testul stă la final.

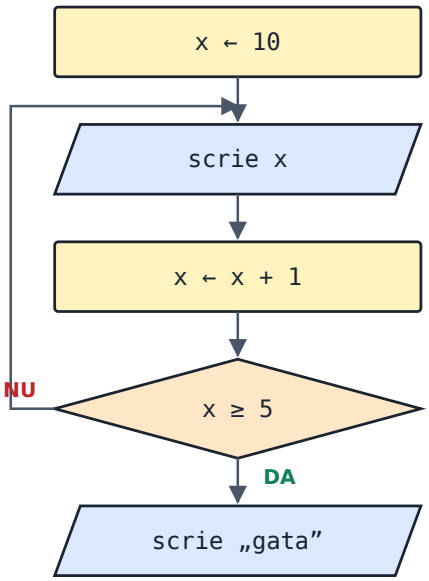
2. Diferența față de „cât timp”

	`cât timp C execută`	`repetă … până când C`
unde e testul	la început	la final
de câte ori se execută minim	0	1
când se oprește	când C devine **fals**	când C devine **adevărat**

Aceleași date (x = 10), comportament diferit:



cât timp: 0 repetări



repetă până când: 1 repetare

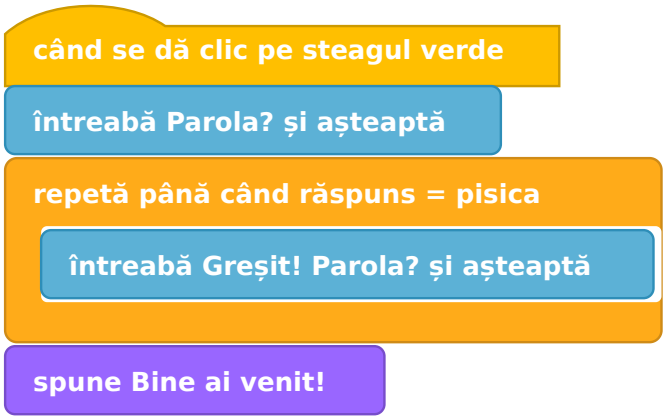
Comparație

Structura	**x = 10**	**Afișează**
`cât timp x < 5`	test: 10 < 5 fals → stop	nimic (doar „gata”)
`repetă … până când x ≥ 5`	scrie 10, x = 11; test: 11 ≥ 5 adevărat → stop	10

3. În Scratch

Scratch nu are un bloc cu test la final, dar îl putem imita: facem pașii **o dată**, apoi folosim `repetă până când`.

Exemplu: cere o parolă până e corectă.



Parola

În pseudocod, mai natural:

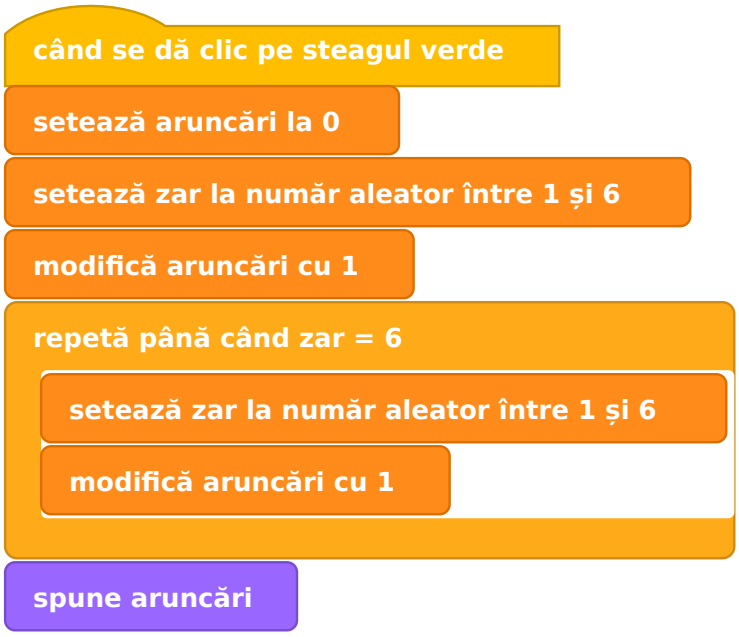
```
repetă
    citește parola
până când parola = "pisica"
scrie "Bine ai venit!"
```

DE CE AȘA?

De ce în Scratch scriem întrebarea de două ori? Pentru că `repetă până când` testează **înainte**. Ca să cerem parola măcar o dată, o cerem o dată înainte de bloc; în interior o cerem din nou doar dacă a fost greșită.

4. Exemplu: zarul

Aruncăm un zar **până iese 6** și numărăm aruncările. Cel puțin o aruncare se face sigur: test final.



Zarul

Un exemplu de rulare (zarul este aleator, deci altă dată iese altceva):

	zar	aruncări	zar = 6?
prima aruncare	3	1	nu
a 2-a	5	2	nu
a 3-a	2	3	nu
a 4-a	6	4	**da → stop**

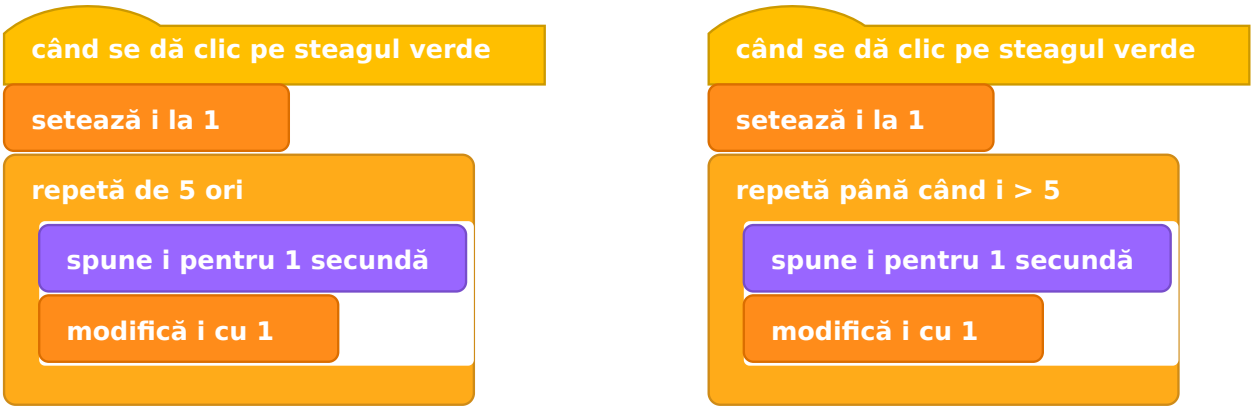
programul spune 4; la altă rulare iese alt număr: zarul e aleator

Urmărire

5. Cele trei structuri repetitive, pe scurt

Structura	Când o folosim	Exemplu
`pentru i ← 1, n`	știm de câte ori	desen poligon, numărătoare
`cât timp`	nu știm; se poate să nu repetăm deloc	cifrele unui număr
`repetă … până când`	nu știm; pașii trebuie făcuți măcar o dată	parola, zarul

Numerele de la 1 la 5, în Scratch, în două feluri:



Două feluri

6. Greșeli frecvente

Greșeala	Efect	Corect
confunzi „oprește când C devine adevărat” cu „continuă cât C e adevărat”	repetă invers	`repetă … până când` = oprire la C adevărat
uiți să modifici variabila din condiție	buclă infinită	schimbă ceva în interior
faci testul înainte când ai nevoie de el după	prima valoare ratată	folosește test final
în Scratch uiți prima rulare înainte de bloc	întrebarea nu se pune niciodată	pașii o dată + `repetă până când`
condiția opusă (`x < 5` în loc de `x ≥ 5`)	repetă greșit	scrie și opusul pe hârtie

7. Exersează pe hârtie

1. Ce afișează? `x ← 1`, `repetă scrie x, x ← x * 2 până când x > 20`.

2. Câte repetări are `x ← 0`, `repetă x ← x + 3 până când x > 10`? Ce valoare are x la final?

3. Scrie algoritmul „afișează numerele de la 1 la 5" în toate cele trei feluri (pentru, cât timp, repetă ... până când).

4. Transformă `cât timp x < 50 execută x ← x * 2` în `repetă … până când`. Sunt echivalente pentru x = 60? Dar pentru x = 3?

5. Pentru fiecare situație alege: cât timp sau repetă ... până când: a) citești un număr până când e pozitiv; b) numeri cifrele unui număr dat; c) arunci zarul până iese 6.

6. În jocul zarului, aruncările au fost 2, 5, 1, 6. Câte aruncări numără programul? Completează tabelul.

7. Ce afișează `n ← 5`, `repetă scrie n, n ← n − 2 până când n < 0`?

8. Scrie în pseudocod: „citește numere până când primești 0, apoi afișează câte numere au fost citite (fără 0)".

9. Care este condiția de oprire pentru `repetă … până când` dacă vrei să te oprești când `x ≥ 10`? Dar la `cât timp` pentru același algoritm?

10. De ce nu există „repetă ... până când" cu 0 repetări?

8. Provocări în Scratch

1. Zarul se aruncă (`număr aleator între 1 și 6`) **până când** iese 6. Numără câte aruncări au fost.
2. Parola: cere „pisica" până o scrie corect; după 3 greșeli spune „Încearcă mâine!".
3. „Ghicește numărul" completat: mai mare / mai mic / corect, cu numărul de încercări.
4. **Dificil:** cere numere până primește 0; spune suma lor.

9. Mini-grile

1. `repetă … până când C` se execută: a) de cel puțin 0 ori b) de cel puțin 1 dată c) de exact 1 dată d) niciodată

2. Repetiția `repetă … până când x > 5` se oprește când: a) x ≤ 5 b) x > 5 c) x = 5 d) după 5 pași

3. Pentru o parolă folosim: a) pentru b) cât timp, fără pași înainte c) repetă ... până când d) nicio repetiție

4. Ce afișează `x ← 1`, `repetă scrie x, x ← x * 2 până când x > 20`? a) 1 2 4 8 16 b) 1 2 4 8 16 32 c) 2 4 8 16 d) 1 2 4

10. Mai departe (pentru curioși)

`do … while` în C++. În limbajele reale, „repetă ... până când" se numește `do … while`, iar condiția este în sens invers: continuă **cât timp** e adevărată. Pseudocodul `repetă … până când x ≥ 10` devine `do { … } while (x < 10);`.

Meniuri. Programele reale cer o opțiune „până când" utilizatorul alege una validă: exact structura cu test final.

Verificare de date. „Citește vârsta până când e între 0 și 120" este o aplicație tipică: nu poți verifica vârsta înainte s-o citești.

11. Ce am învățat

- `repetă … până când` face pașii cel puțin o dată și se oprește când condiția e adevărată.
- `cât timp` poate să nu repete deloc; `repetă … până când` repetă măcar o dată.
- Alegem structura în funcție de ce știm despre numărul de repetări.
- În Scratch imităm testul final: pașii o dată, apoi `repetă până când`.

Citește și exersează în plus

- Scratch — editorul online — aici rulezi exemplele din lecție https://scratch.mit.edu/projects/editor/
- Scratch Wiki — Control blocks — repetările în Scratch: repetă, la nesfârșit, repetă până când (engleză) https://en.scratch-wiki.info/wiki/Control_Blocks
- Code.org — Hour of Code — puzzle-uri cu repetiții și condiții, gratuite https://code.org/hourofcode/overview

Răspunsuri

Exersează pe hârtie

1. x = 1, 2, 4, 8, 16, 32: se afișează 1, 2, 4, 8, 16 (după 16, x devine 32 > 20 și se oprește fără să afișeze 32). Afișează **1 2 4 8 16**.

2. x = 3, 6, 9, 12 (12 > 10: stop). **4** repetări; x = **12**.

3. Pentru: `pentru i ← 1, 5 execută scrie i`. Cât timp: `i ← 1`, `cât timp i ≤ 5 execută scrie i, i ← i + 1`. Repetă: `i ← 1`, `repetă scrie i, i ← i + 1 până când i > 5`.

4. `repetă x ← x * 2 până când x ≥ 50`. Pentru x = 60: cât timp nu face nimic (x rămâne 60), repetă face o dată (x = 120): **nu sunt echivalente**. Pentru x = 3 rezultă același x final (96).

5. a) repetă ... până când (citirea se face măcar o dată); b) cât timp (numărul poate fi 0 sau mic); c) repetă ... până când (cel puțin o aruncare).

6. 4 aruncări (2, 5, 1, 6), numărul de aruncări crește la fiecare, deci la final aruncări = **4**.

7. n = 5, 3, 1, −1: se afișează 5, 3, 1; după 1, n devine −1 < 0 și se oprește. **5 3 1**.

8. `k ← 0` / `citește x` / `cât timp x ≠ 0 execută k ← k + 1, citește x` / `scrie k` (sau: `repetă citește x, dacă x ≠ 0 atunci k ← k + 1 până când x = 0`).

9. La „repetă ... până când": `x ≥ 10`. La „cât timp": opusul, `x < 10`.

10. Pentru că pașii se execută **înainte** de test: măcar o dată.

Mini-grile

1. b. **2.** b. **3.** c. **4.** a.