INFORMATICĂ · CLASA A VI-A · ALGORITMI · LECȚIA 1

Recapitulare și reprezentarea algoritmilor: scheme logice și pseudocod

Durată: 45 min Nivel: gimnaziu Înainte: lecțiile g5-05

CE VEI ȘTI DUPĂ LECȚIE

- Îți amintești structurile secvențială și alternativă.
- Desenezi o schemă logică simplă cu blocurile standard.
- Scrii același algoritm în pseudocod, respectând indentarea.
- Traduci un algoritm din Scratch în pseudocod și invers.

1. Ce știm din clasa a V-a

- **Algoritm** = pași clari, în ordine, care se termină și dau un rezultat.
- **Structura secvențială** = pașii unul după altul.
- **Structura alternativă** = `dacă … atunci … altfel`.
- **Variabile** (cutii cu nume) și **expresii** (aritmetice, relaționale, logice).

Azi învățăm să **reprezentăm** un algoritm în două feluri noi, mai exacte decât cuvintele obișnuite: **schema logică** (desen) și **pseudocodul** (text scurt, ca un cod simplificat).

2. Schema logică

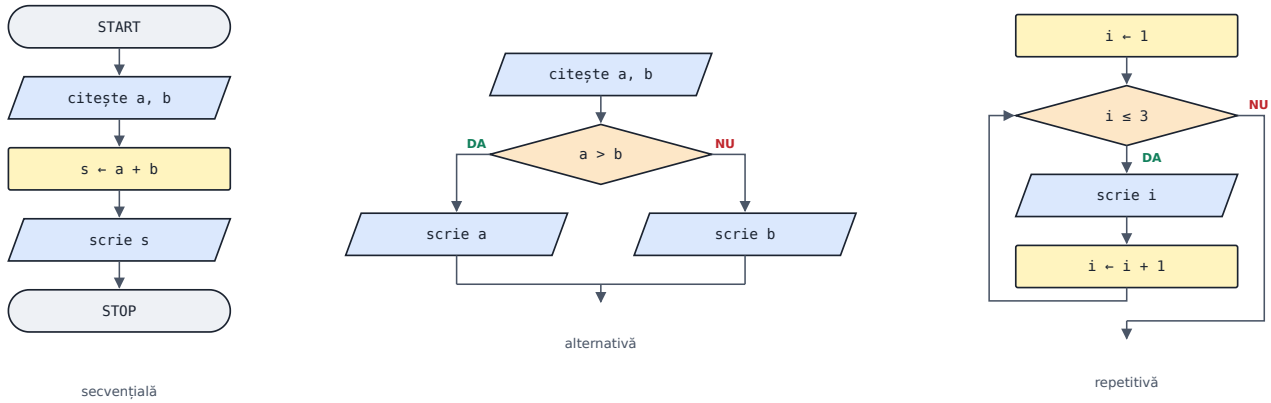
Un desen al algoritmului, cu forme standard unite prin săgeți:



Simboluri

Formă	Rol
oval	START / STOP (începutul și sfârșitul)
paralelogram	citire sau afișare („citește n”, „scrie s”)
dreptunghi	calcul („s ← a + b”)
romb	decizie (o întrebare cu răspuns DA / NU)

Reguli: o schemă are **un singur** START; fiecare formă are o săgeată care intră și (în afară de STOP și de romb) una care iese; rombul are **două** ieșiri, DA și NU.



Cele trei structuri

3. Pseudocodul

Este algoritmul scris în cuvinte, într-o formă apropiată de limbajele de programare:

```
citește a, b
dacă a > b atunci
    scrie a
altfel
    scrie b
sfârșit dacă
```

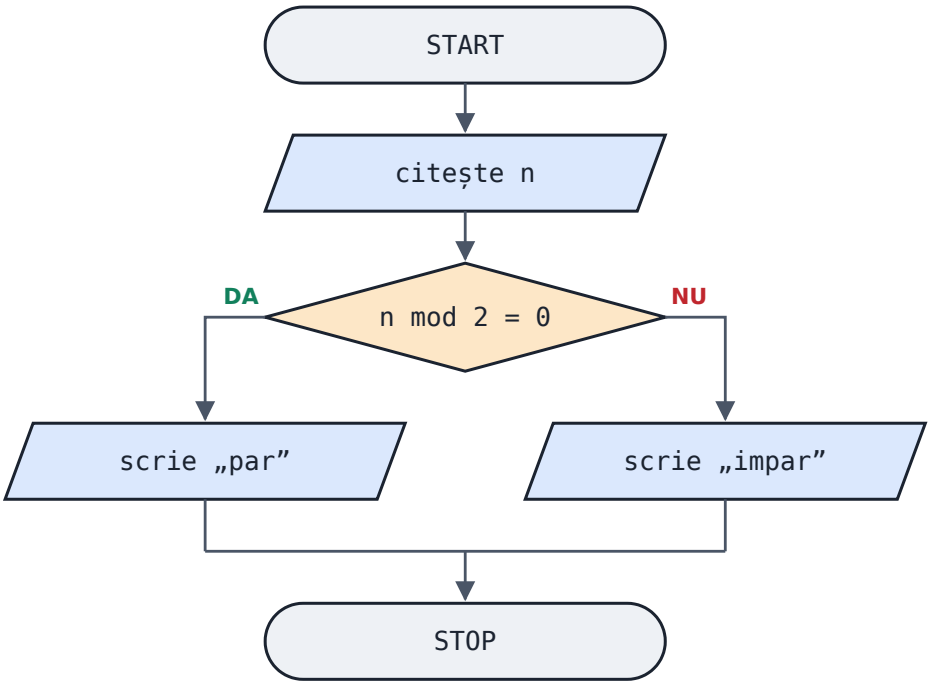
Reguli:

- **săgeata** `←` înseamnă „primește valoarea”: `s ← a + b`;
- ce e „în interiorul” unei structuri se scrie **mai la dreapta** (indentare);
- numele variabilelor sunt scurte și sugestive;
- fiecare structură se închide: `sfârșit dacă`.

> **DE CE AȘA?**
>
> **De ce `←` și nu `=`?** În matematică `s = a + b` este o egalitate. În algoritmi `s ← a + b` este o **acțiune**: calculează a + b și pune rezultatul în cutia `s`. Din acest motiv, `x ← x + 1` are sens în algoritm (mărește x cu 1), dar nu are sens ca egalitate. Iar `=` îl păstrăm pentru **a întreba** dacă două valori sunt egale (`dacă x = 5`).

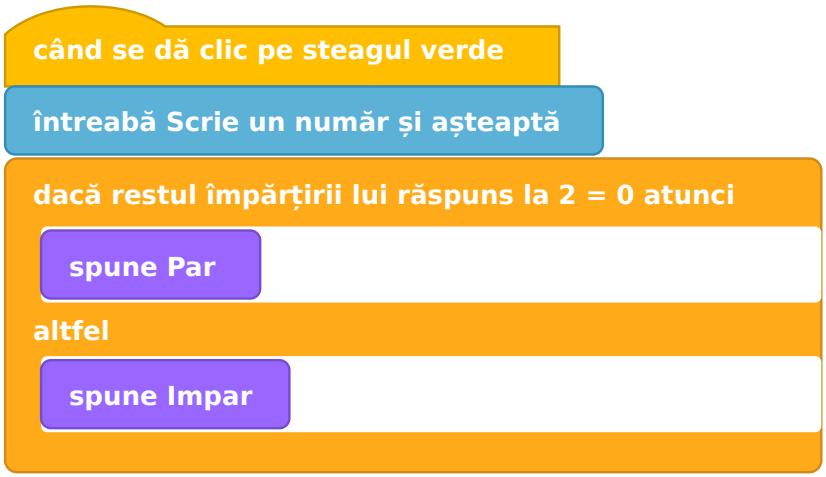
4. Același algoritm, trei feluri

Algoritmul „par sau impar”:



n mod 2 = restul împărțirii lui n la 2

Schema



Scratch

```
citește n
dacă n mod 2 = 0 atunci
    scrie "par"
altfel
    scrie "impar"
sfârșit dacă
```

Schema logică	Scratch	Pseudocod
paralelogram „citește n”	`întreabă … și așteaptă`	`citește n`

Schema logică	Scratch	Pseudocod
romb cu două ieșiri	`dacă … atunci … altfel`	`dacă … atunci … altfel`
paralelogram „scrie”	`spune …`	`scrie …`
oval START / STOP	`când se dă clic pe steag`	început / sfârșit

Spune-le „traducere”: cunoști un singur algoritm, dar îl poți scrie în trei „limbi”.

5. Urmărim un algoritm pe hârtie

```
citește a, b
s ← a + b
dacă s > 10 atunci
    scrie s - 10
altfel
    scrie s
sfârșit dacă
```

	s	s > 10?	scrie
a = 3, b = 8	11	da (11 > 10)	**1**
a = 9, b = 2	11	da (11 > 10)	**1**
a = 2, b = 3	5	nu	**5**

două perechi diferite pot da același rezultat

Urmărire

DE CE AȘA?

De ce facem tabelul? Calculatorul execută exact; noi, uitându-ne rapid, putem greși. Tabelul pe hârtie arată valorile fiecărei variabile la fiecare pas și ne spune ce ramură se alege. Așa verificăm un algoritm **înainte** să-l programăm.

6. Greșeli frecvente

Greșeala	Cum se repară
romb cu o singură ieșire	rombul are mereu DA și NU
săgeți care se încrucișează sau lipsesc	desenează pe rând, cu rigla
`a = b` în loc de `a ← b` pentru atribuire	`←` pentru „primește”, `=` pentru „este egal?”
uiți `sfârșit dacă`	închide fiecare structură
nu scrii START / STOP	oricare schemă are început și sfârșit
uiți indentarea	ce e în interior se scrie mai la dreapta

7. Exersează pe hârtie

1. Desenează schema logică și scrie pseudocodul pentru: „citește un număr; dacă e pozitiv scrie POZITIV, altfel scrie NEGATIV SAU ZERO”.

2. Ce afișează algoritmul din secțiunea 5 pentru a = 2, b = 3? Dar pentru a = 20, b = 5?

3. Transformă în pseudocod programul Scratch „par sau impar” (a fost dat în lecție). Scrie-l apoi pentru „multiplu de 3”.

4. Ce valoare are `x` la final? `x ← 4`, `x ← x + 3`, `x ← x * 2`.

5. Scrie pseudocodul pentru „citește două numere și afișează suma lor”. Desenează schema.

6. Scrie pseudocodul pentru „citește vârsta; dacă este cel puțin 18, scrie MAJOR, altfel MINOR”.

7. Care este greșeala? `citește a` / `dacă a > 5 atunci` / `scrie a` / (nu există „altfel” sau „sfârșit dacă”). Ce lipsește?

8. Desenează schema pentru algoritmul care scrie cel mai mic dintre două numere.

9. Ce afișează: `citește n`, `dacă n < 0 atunci n ← 0 − n sfârșit dacă`, `scrie n` pentru n = −7 și n = 5? (Ce face algoritmul?)

10. Scrie algoritmul „schimbă între ele două variabile a și b” (indiciu: ai nevoie de o a treia cutie).

8. Mini-grile

1. Ce formă se folosește pentru o decizie? a) oval b) paralelogram c) dreptunghi d) romb

2. `x ← x + 1` înseamnă: a) x este egal cu x + 1 b) mărește x cu 1 c) x devine 0 d) eroare

3. Ce formă se folosește pentru „scrie s”? a) oval b) paralelogram c) romb d) dreptunghi

4. Câte ieșiri are un romb? a) una b) două c) trei d) niciuna

9. Mai departe (pentru curioși)

Aproape cod real. Pseudocodul din lecție se transformă ușor într-un limbaj de programare. `dacă n mod 2 = 0 atunci` devine în Python `if n % 2 == 0:` și în C++ `if (n % 2 == 0)`. Observi că `=` din pseudocod devine `==` (egalitate), iar `←` devine `=` (atribuire): exact inversul!

Scheme logice pe calculator. Programele gratuite de desen (draw.io, Flowgorithm) desenează schema și o execută pas cu pas.

Verificarea cu valori „la limită”. Un algoritm bun se testează cu valori speciale: 0, numere negative, numere egale. Pentru algoritmul „cel mai mic dintre două numere”, încearcă și a = b.

10. Ce am învățat

- Schema logică: oval, paralelogram, dreptunghi, romb.
- Pseudocodul: cuvinte + `←` + indentare.
- Același algoritm se poate scrie în Scratch, în schemă logică sau în pseudocod.
- Urmărim algoritmul într-un tabel înainte să-l programăm.

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

Răspunsuri

Exersează pe hârtie

1. Schema: START → paralelogram „citește n” → romb „n > 0?” → DA: „scrie POZITIV”, NU: „scrie NEGATIV SAU ZERO” → STOP. Pseudocod: `citește n` / `dacă n > 0 atunci scrie "POZITIV" altfel scrie "NEGATIV SAU ZERO" sfârșit dacă`.

2. a = 2, b = 3: s = 5, nu e > 10, afișează **5**. a = 20, b = 5: s = 25 > 10, afișează 25 − 10 = **15**.

3. Pentru „multiplu de 3”: `citește n` / `dacă n mod 3 = 0 atunci scrie "multiplu" altfel scrie "nu" sfârșit dacă`.

4. 4 → 7 → **14**.

5. `citește a, b` / `s ← a + b` / `scrie s`. Schema: START → „citește a, b” → [s ← a + b] → „scrie s” → STOP.

6. `citește v` / `dacă v ≥ 18 atunci scrie "MAJOR" altfel scrie "MINOR" sfârșit dacă`.

7. Lipsește `sfârșit dacă` (structura trebuie închisă). „altfel” poate lipsi (decizie fără ramura NU), dar atunci nu se scrie nimic dacă a ≤ 5.

8. START → „citește a, b” → romb „a < b?” → DA: „scrie a”, NU: „scrie b” → STOP. (Pentru a = b scrie b: valoarea e aceeași.)

9. n = −7: −7 < 0, n ← 0 − (−7) = 7, scrie **7**. n = 5: nu intră în decizie, scrie **5**. Algoritmul afișează **valoarea absolută** a lui n.

10. `c ← a`, `a ← b`, `b ← c`.

Mini-grile

1. d. **2.** b. **3.** b. **4.** b.