FIȘĂ DE LUCRU · CLASA A IX-A · FIȘA 2 · ACOPERĂ LECȚIA IX.4

Decizii: if, else if, switch

Durată: 90 min Nivel: toate profilurile

CE VEI ȘTI DUPĂ LECȚIE

- Prezici ce ramură se execută într-un lanț else if, în if-uri imbricate și într-un switch fără break.
- Repari greșelile clasice: ordinea ramurilor, ; după if, acolade lipsă, break uitat.
- Scrii decizii pentru probleme reale și alegi testele care verifică fiecare ramură și fiecare capăt.

Ce trebuie să știi (o pagină)

Regula	Exemplu	Ce se întâmplă
`if` / `else`: se execută **exact o** ramură	`if (n % 2 == 0) ... else ...`	par sau impar, niciodată ambele
lanț `else if`: câștigă **prima** condiție adevărată	`if (p >= 90) ... else if (p >= 75) ...`	restul nu se mai testează; ordinea contează
`if`-uri **separate**: fiecare se testează	`if (a) ...; if (b) ...;`	se pot executa mai multe
fără acolade, `if` ia **o singură** instrucțiune	`if (c) x = 1; y = 2;`	`y = 2` se execută mereu
`if (c);` are corpul **gol**	`if (x > 0); { ... }`	blocul rulează mereu
`else` se leagă de **cel mai apropiat** `if` fără `else`	`if (a) if (b) X; else Y;`	`Y` ține de `if (b)`, indiferent de indentare
`switch` merge pe `int` / `char`; `case` = punct de **intrare**	`case 'a': case 'e': cout << "vocala"; break;`	fără `break`, „cade” în cazul următor
scurtcircuit: în `A && B`, dacă `A` e fals, `B` nu se calculează	`if (b != 0 && a / b > 2)`	nu se împarte la 0
o comparație valorează 0 sau 1	`(a == b) + (b == c)`	numără egalitățile
`cond ? x : y`	`cout << (n < 0 ? -n : n);`	modulul lui `n`

SFAT

Un program cu decizii e testat abia când ai trecut **o dată prin fiecare ramură** și ai încercat **fiecare capăt** de interval (`17` și `18`, nu doar `10` și `30`).

Nivelul 1 · Prezice

★ citește și prezice

P1. Cele două secvențe arată aproape la fel. Completează tabelul cu ce afișează fiecare.

```cpp
// A                                    // B
if (t < 0) cout << "G";                 if (t < 0) cout << "G";
else if (t < 15) cout << "F";           if (t < 15) cout << "F";
else if (t < 25) cout << "P";           if (t < 25) cout << "P";
else cout << "C";                       else cout << "C";
```

t	−5	10	15	30
A				
B				

P2. Indentarea minte. Ce afișează pentru `x = 5, y = -2`? Dar pentru `x = -1, y = 3`?

```cpp
if (x > 0)
    if (y > 0) cout << "A";
else cout << "B";
cout << "C";
```

P3. Ce afișează? A reușit programul să interschimbe valorile?

```cpp
int a = 3, b = 7, aux = 0;
if (a > b)
    aux = a; a = b; b = aux;
cout << a << ' ' << b;
```

P4. Completează ce se afișează pentru `k = 4, 5, 6, 7`.

```cpp
switch (k % 4)
{
    case 0: cout << "N";
    case 1: cout << "E"; break;
    case 2: cout << "S";
    default: cout << "V";
}
```

P5. Ce afișează pentru `a = 0`? Dar pentru `a = -2`? (Urmărește când se execută `++b`.)

```cpp
int b = 10, c = 0;
if (a != 0 && b / a > 2) c = 1;
else if (a == 0 || ++b > 0) c = 2;
cout << b << ' ' << c;
```

P6. Ce valoare are `m` pentru `(a, b, c) = (5, 2, 8)`, `(1, 9, 4)` și `(7, 7, 3)`? Spune **în cuvinte** ce calculează secvența.

```cpp
if (a > b) { if (b > c) m = b; else if (a > c) m = c; else m = a; }
else       { if (a > c) m = a; else if (b > c) m = c; else m = b; }
```

P7. *Grilă de BAC.* Care expresii au valoarea 1 dacă și numai dacă `x` aparține mulțimii $[-3, 2) \cup [5, 9]$? (Pot fi mai multe corecte. Metodă: testează capetele −3, 2, 5, 9.)

a. `x >= -3 && x < 2 || x >= 5 && x <= 9`

b. `!(x < -3 || x >= 2) || !(x < 5 || x > 9)`

c. `x >= -3 && x <= 9 && !(x >= 2 && x <= 5)`

d. `x > -3 && (x < 2 || x >= 5) && x <= 9`

P8. Ce afișează pentru `4 4 4`, `4 9 4`, `1 2 3`? Poate ajunge programul vreodată pe ramura `default`? De ce?

```
int k = (a == b) + (b == c) + (a == c);
switch (k)
{
    case 3: cout << "toate egale"; break;
    case 1: cout << "doua egale"; break;
    case 0: cout << "distincte"; break;
    default: cout << "imposibil";
}
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. FizzBuzz: pentru multiplii lui 15 trebuie afișat `FizzBuzz`. Ce afișează programul pentru `n = 30`? Care ramură nu se execută **niciodată**? Repară mutând un singur rând.

```
if (n % 3 == 0) cout << "Fizz";
else if (n % 5 == 0) cout << "Buzz";
else if (n % 15 == 0) cout << "FizzBuzz";
else cout << n;
```

P10. Cardul de autobuz are `credit` lei, iar o călătorie costă 3 lei. Ce afișează pentru `credit = 1`? Găsește greșeala.

```
if (credit >= 3);
{
    credit = credit - 3;
    cout << "Calatorie platita. ";
}
cout << "Credit ramas: " << credit;
```

P11. Secvența trebuie să dea numărul de zile ale lunii (an nebisect), dar dă 31 pentru **orice** lună. De ce? Repară-o fără să ștergi niciun `case`.

```
switch (luna)
{
    case 2: zile = 28;
    case 4: case 6: case 9: case 11: zile = 30;
    default: zile = 31;
}
```

P12. *Puzzle cu rânduri amestecate.* Programul rezolvă ecuația `a·x + b = 0` și afișează soluția, `infinitate` sau `nu are solutii`. Pune rândurile în ordine; un rând este în plus.

```
A)  else cout << (double)-b / a;
B)  if (a == 0)
C)  {   if (b == 0) cout << "infinitate";
D)      else cout << "nu are solutii";  }
E)  cin >> a >> b;
F)  else cout << -b / a;
G)  int a, b;
```

P13. Programul trebuie să numere câte dintre `a`, `b`, `c` sunt pare. Găsește un test pe care merge și unul pe care greșește. Ce greșeală a făcut elevul?

```
int cnt = 0;
if (a % 2 == 0) cnt++;
else if (b % 2 == 0) cnt++;
else if (c % 2 == 0) cnt++;
```

P14. *Completează golurile.* Laturile sunt citite deja **ordonate**: `a ≤ b ≤ c`. Folosește faptul acesta ca să scrii condiții cât mai scurte.

```
if (__________) cout << "nu e triunghi";
else if (__________) cout << "echilateral";
else if (__________) cout << "isoscel";
else cout << "oarecare";
```

Teste: `3 4 5 → oarecare`, `2 2 3 → isoscel`, `1 2 3 → nu e triunghi`, `5 5 5 → echilateral`.

Nivelul 3 · Scrie cod

★★ scrie cod

Pentru fiecare program, verifică **toate** testele date înainte să spui că e gata.

P15. Parcarea. Se citește durata `m` în minute (0 ≤ m ≤ 1440). Prima oră e gratuită; după ea se plătesc 5 lei pentru **fiecare oră începută**, dar cel mult 40 de lei pe zi.

`m`	45	60	61	150	600
plată	`0`	`0`	`5`	`10`	`40`

P16. Piatră, hârtie, foarfecă. Se citesc două caractere din `P`, `H`, `F` (alegerile celor doi jucători). Piatra bate foarfeca, foarfeca bate hârtia, hârtia bate piatra. Afișează `jucatorul 1`, `jucatorul 2` sau `egal`.

intrare	`P F`	`P H`	`F F`	`H P`	`F P`
ieșire	`jucatorul 1`	`jucatorul 2`	`egal`	`jucatorul 1`	`jucatorul 2`

P17. Tastatura telefonului. Pe tastele vechilor telefoane: `abc`→2, `def`→3, `ghi`→4, `jkl`→5, `mno`→6, `pqrs`→7, `tuv`→8, `wxyz`→9. Citește o literă mică și afișează tasta, folosind `switch` cu **cazuri grupate**. Teste: `a → 2`, `s → 7`, `z → 9`, `o → 6`, `v → 8`.

P18. Ziua de mâine. Se citește o dată validă `zi luna` dintr-un an nebisect. Afișează data zilei următoare (refolosește `switch`-ul reparat de la P11).

intrare	`28 2`	`31 12`	`30 4`	`15 7`	`30 1`
ieșire	`1 3`	`1 1`	`1 5`	`16 7`	`31 1`

P19. Podiumul. Trei alergători au timpii `t1`, `t2`, `t3` (distincți; timpul mai mic e mai bun). Afișează **numerele de concurs** ale alergătorilor în ordinea podiumului: locul 1, 2, 3.

intrare	`12 9 15`	`30 20 10`	`5 6 7`	`8 3 5`
ieșire	`2 1 3`	`3 2 1`	`1 2 3`	`2 3 1`

P20. Programări suprapuse. Două ședințe ocupă intervalele de ore `[a, b]` și `[c, d]` (`a ≤ b`, `c ≤ d`). Afișează intervalul comun sau `vida`.

intrare	`1 5 3 9`	`1 2 3 4`	`2 8 3 4`	`1 3 3 7`
ieșire	`3 5`	`vida`	`3 4`	`3 3`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Corect sau nu? Ambele variante vor să pună în `mx` maximul lui `a`, `b`, `c`. Una este corectă pentru **orice** valori, cealaltă nu. Care e care? Pentru cea greșită dă un contraexemplu; pentru cea corectă explică de ce ajunge, pe ramura `else if`, să compari doar `b` cu `c`.

```
// V1
if (a > b && a > c) mx = a;
else if (b > c) mx = b;
else mx = c;
```

```
// V2
mx = 0;
if (a > b && a > c) mx = a;
if (b > a && b > c) mx = b;
if (c > a && c > b) mx = c;
```

P22. Parcarea fără `if`. Rescrie calculul de la P15 într-o singură expresie, fără `if` și fără `?:` (ai voie la `min`). Indiciu: câte ore începute sunt după prima oră? Explică de ce `m = 0` nu strică formula.

P23. Cât mai puține comparații. Ca să afli maximul **și** minimul a 4 numere, varianta directă face 3 + 3 = 6 comparații. Se poate cu doar **4**. Cum? (Indiciu: compară-le întâi pe perechi.)

P24. Testele minime. Biletul la cinema: sub 7 ani e gratuit, de la 7 la 17 ani și de la 65 de ani în sus costă 15 lei, altfel 30 de lei. Alege **cel mai mic** set de vârste care verifică fiecare ramură și fiecare capăt de interval. Ce preț trebuie să dea programul pentru fiecare?

P25. Mini-proiect: ghicitorul. Scrie un program care ghicește unul dintre **8** animale la care se gândește utilizatorul, punând întrebări cu răspuns `d` / `n` (de exemplu „Are blană?”). Desenează întâi arborele de întrebări. Câte întrebări pune programul, cel mult? Câte `if`-uri ai scris în total? De ce nu se poate cu mai puține întrebări?

Probleme de lucru în clasă (pbinfo)

Le rezolvăm împreună în ședință, în ordinea de mai jos.

- [ ] `#105` **max2** — ușoară
  https://www.pbinfo.ro/probleme/105/max2

- [ ] #9 **maxmin** — ușoară
  https://www.pbinfo.ro/probleme/9/maxmin
- [ ] #469 **interval2** — ușoară
  https://www.pbinfo.ro/probleme/469/interval2
- [ ] #1303 **calculator** — ușoară
  https://www.pbinfo.ro/probleme/1303/calculator

Temă pentru acasă (pbinfo)

Trimite soluțiile pe pbinfo și notează punctajul obținut. Bifează ce ai terminat.

- [ ] #179 **triunghi** — ușoară
  https://www.pbinfo.ro/probleme/179/triunghi
- [ ] #453 **distincte1** — ușoară
  https://www.pbinfo.ro/probleme/453/distincte1
- [ ] #832 **nota** — ușoară
  https://www.pbinfo.ro/probleme/832/nota
- [ ] #1625 **ec2** — ușoară
  https://www.pbinfo.ro/probleme/1625/ec2