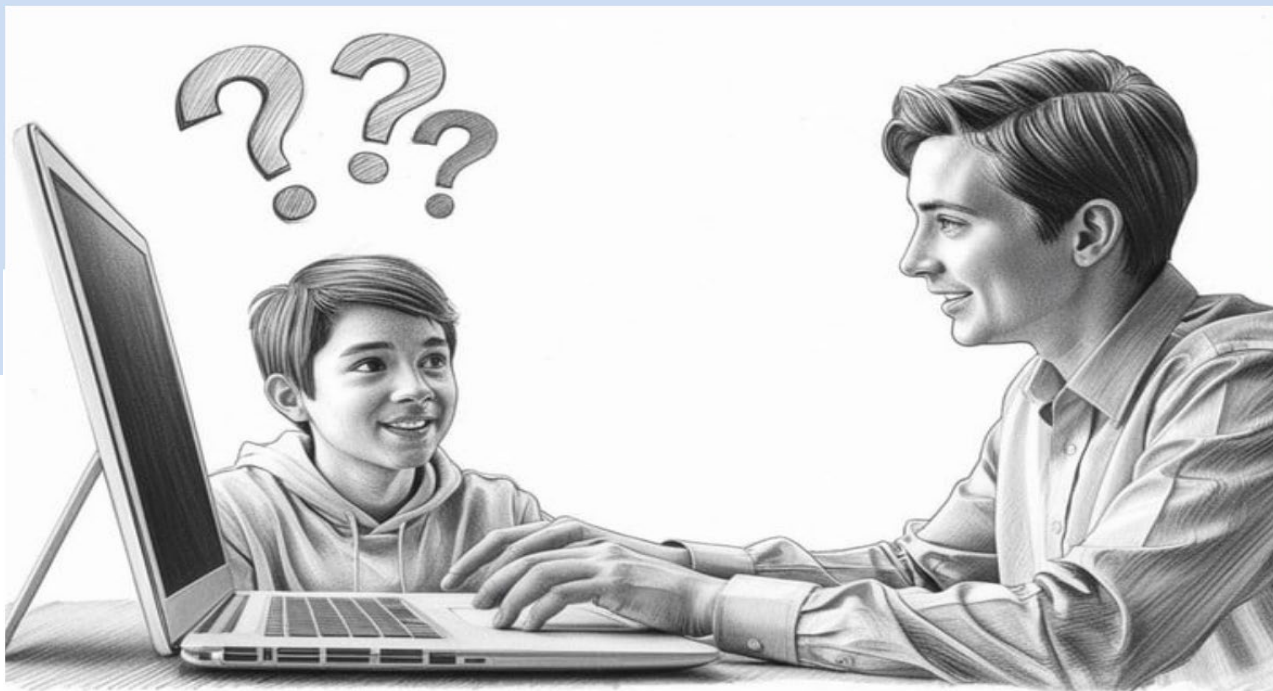


GRIBINEȚ LUMINIȚA

MUSTEAȚĂ VICTORIA

# **Integrarea metodei problematizării în predarea informaticii pentru dezvoltarea competențelor algoritmice la elevi**

*Broșură metodică pentru profesorii debutanți*



Această resursă metodologică a fost concepută ca un instrument de formare și ghidare pentru cadrele didactice aflate la început de drum în cariera pedagogică, în special pentru profesorii debutanți din cadrul Centrului de Excelență în Informatică și Tehnologii Informaționale, precum și pentru tinerii specialiști din întregul sistem de învățământ profesional tehnic. Acest ghid pentru profesorii debutanți a fost analizat și aprobat în cadrul ședinței Catedrei Informatică II, Proces-verbal nr. 2 din 20.10.2021 și ulterior aprobată de Consiliul Metodico-Științific al CEITI, Proces-verbal nr. 3 din 04.11.2021. Recomandată ca sprijin real pentru tinerii specialiști, dar și pentru cadrele didactice care revin în sistemul educațional după o pauză profesională, oferindu-le o bază actualizată și aplicabilă pentru reconectarea la realitățile și cerințele actuale ale predării în domeniul TIC.

Broșura metodică este rezultatul unor experiențe validate în practică și reflectă efortul constant al autoarelor, desfășurat pe parcursul mai multor ani de implementare a acestui model metodologic în cadrul activităților didactice și sesiunilor de pregătire a elevilor pentru Olimpiade și alte concursuri naționale și internaționale. Lucrarea de față este orientată spre îmbinarea riguroasă a teoriei pedagogice cu realitățile concrete ale predării disciplinelor din domeniul TIC în instituțiile de învățământ profesional tehnic din Republica Moldova.

#### Autori:

Gribineț Luminița, profesoară la disciplinele de informatică, grad didactic unu

Musteață Victoria, profesoară la disciplinele de informatică, grad didactic doi

#### Coordonator:

Jumbei Olga, Magistră în Științe ale Educației, metodistă și profesoară, grad didactic superior

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| Prefață .....                                                                    | 3  |
| Prezentare generală .....                                                        | 4  |
| I. Fundamente pedagogice .....                                                   | 5  |
| 1.1 Ce este metoda problematizării? .....                                        | 5  |
| 1.2 De ce este relevantă în predarea informaticii? .....                         | 5  |
| 1.3 Ce spune teoria educațională? .....                                          | 5  |
| 1.4 Ce înseamnă să predai prin problematizare? .....                             | 6  |
| II. Etapele de implementare .....                                                | 6  |
| 2.1 Planificarea didactică și integrarea metodei .....                           | 6  |
| 2.2 Crearea resurselor educaționale .....                                        | 7  |
| 2.3 Desfășurarea lecțiilor interactive .....                                     | 8  |
| 2.4 Evaluarea învățării prin problematizare .....                                | 8  |
| 2.5 Monitorizarea și adaptarea continuă .....                                    | 9  |
| III. Rolul profesorului în predarea prin problematizare .....                    | 9  |
| 3.1 Profesorul ca facilitator .....                                              | 10 |
| 3.2 Construirea unui climat de învățare deschis .....                            | 10 |
| 3.3 Ghid pentru formularea întrebărilor provocatoare .....                       | 10 |
| IV. Resurse utile pentru aplicare .....                                          | 11 |
| 4.1 Fișe metodologice (RED) .....                                                | 11 |
| 4.2 Setul de planșe didactice aprobate .....                                     | 11 |
| 4.3 Exemple de sarcini și activități .....                                       | 12 |
| V. Recomandări pentru profesorii debutanti .....                                 | 12 |
| 5.1 Sfaturi practice pentru activități didactice .....                           | 12 |
| 5.2 Erori frecvente și cum le evităm .....                                       | 13 |
| 5.3 Când și cum adaptăm metoda în funcție de stilul de învățare a elevilor ..... | 13 |
| Concluzii .....                                                                  | 14 |
| Referințe bibliografice .....                                                    | 15 |
| Anexa 1 .....                                                                    | 16 |
| Anexa 2 .....                                                                    | 17 |
| Anexa 3 .....                                                                    | 18 |
| Anexa 4 .....                                                                    | 21 |
| Anexa 5 .....                                                                    | 22 |
| Anexa 6 .....                                                                    | 25 |
| Anexa 7 .....                                                                    | 27 |
| Anexa 8 .....                                                                    | 29 |
| Anexa 9 .....                                                                    | 30 |
| Anexa 10 .....                                                                   | 32 |

## Prefață

Într-un context educațional aflat în continuă transformare, profesorii debutanți au nevoie, mai mult ca oricând, de repere clare și de instrumente metodologice aplicabile, care să le susțină începutul profesional și să-i încurajeze să dezvolte practici de calitate, încă din primii pași în cariera pedagogică.

Această broșură metodică este construită ca un ghid practic și coerent, menit să sprijine cadrele didactice aflate la început de drum în predarea informaticii, prin integrarea metodei problematizării – o strategie activă, modernă, care pune în centru gândirea algoritmică, logica și capacitatea elevilor de rezolvare aplicativă a problemelor.

Documentul de referință oferă o abordare metodologică clară și structurată, pornind de la fundamentele pedagogice ale metodei problematizării și ajungând până la exemple de planificări, fișe de lucru și sugestii concrete de aplicare la clasă. Este, în același timp, un cadru de reflecție profesională și o invitație la dialog cu tentă metodică între generații de profesori.

Aduc mulțumiri colegelor mele, profesoarelor de disciplinele de informatică Gribineț Luminița și Musteață Victoria, pentru implicarea remarcabilă în elaborarea, testarea și validarea acestui model educațional și pentru deschiderea cu care l-au pus la dispoziția comunității didactice din învățământul profesional tehnic.

Îmi exprim încrederea că această broșură va constitui un sprijin valoros pentru toți cei care doresc să îmbine rigoarea informaticii cu provocarea pedagogică a învățării prin problematizare.

Cu respect profesional, **Olga Jumbei**,  
Magistră în Științe ale Educației și metodistă CEITI

## Prezentare generală

Această broșură metodică este elaborată ca un instrument de sprijin profesional destinat cadrelor didactice debutante din învățământul profesional tehnic, care predau discipline din domeniul Tehnologiilor Informaționale și Comunicaționale (TIC).

Documentul oferă un model pedagogic aplicabil de integrare a metodei problematizării în procesul de predare-învățare-evaluare, cu scopul de a susține formarea competențelor esențiale pentru meseriile viitorului: gândirea algoritmică, logica funcțională, capacitatea de a analiza și de a propune soluții viabile în contexte reale.

Metoda propusă pornește de la premisa că învățarea devine autentică atunci când elevii sunt provocați să gândească activ, să exploreze, să greșească și să decidă — exact așa cum sunt solicitați să o facă în mediul profesional din domeniul TIC. Astfel, lecția nu mai este un spațiu de reproducere, ci devine un laborator de idei și soluții.

Broșura este structurată într-un mod clar și etapizat, astfel încât să poată fi utilizată:

- **în planificarea activităților didactice**, prin integrarea întrebărilor problematizante și a obiectivelor centrate pe competențe;
- **în desfășurarea lecțiilor interactive**, prin aplicarea concretă a metodei, însoțită de fișe, exemple și suport vizual;
- **în reflecția pedagogică de după lecție**, oferind repere pentru autoevaluare, ajustare și perfecționare profesională.

### Conținutul broșurii cuprinde:

- fundamente pedagogice esențiale pentru înțelegerea metodei problematizării în contextul TIC;
- pași concreți de implementare, ușor de urmat de către debutanți;
- resurse aplicabile și exemple reale, testate în practică și adaptabile oricărei grupe sau nivel de pregătire;
- modele de fișe problematizante, rubrici de evaluare și planificări curriculare integrate;
- recomandări metodologice pentru adaptare, diferențiere și dezvoltare continuă.

Broșura este disponibilă **exclusiv în format digital**, cu acces facil pentru profesori, și poate fi utilizată:

- individual, pentru dezvoltare profesională personalizată;
- în cadrul ședințelor metodice ale catedrei;
- sau ca material de formare în cadrul sesiunilor organizate la nivel instituțional sau regional.

Scopul principal este de a oferi **sprijin real și aplicabil** cadrelor didactice debutante, care doresc să își construiască o practică pedagogică modernă, coerentă și orientată spre formarea unor competențe relevante pentru viitorul digital al elevilor din ÎPT.

# I. Fundamente pedagogice

## 1.1 Ce este metoda problematizării?

Metoda problematizării este o strategie didactică activă care pune accentul pe învățarea prin întrebări, provocări și rezolvarea de situații concrete. Aceasta presupune plasarea elevului în centrul procesului educațional și determinarea lui să gândească, să formuleze ipoteze, să analizeze și să caute soluții prin efort propriu.

În locul transmiterii directe de informații, profesorul devine facilitator al descoperirii, stimulând gândirea critică a elevului și capacitatea de analiză.

Această metodă pornește de la o situație-problemă, adică o sarcină care:

- nu are un răspuns imediat evident – *aceasta caracteristică determină elevul să caute activ răspunsuri și să-și pună întrebări relevante;*
- necesită aplicarea cunoștințelor – *astfel, elevul este încurajat să-și folosească bagajul cognitiv pentru a înfrunta provocarea didactică;*
- presupune argumentare și explorare – *prin acest proces, învățarea devine profundă și personalizată, pregătind terenul pentru aplicarea metodei în contexte specifice, cum este informatica.*

Această abordare generală devine cu atât mai valoroasă când este aplicată în contexte educaționale specifice, cum ar fi predarea informaticii, unde rezolvarea de probleme este esențială.

## 1.2 De ce este relevantă în predarea informaticii?

În informatică, mai ales în învățământul profesional tehnic este esențial ca elevii să nu învețe doar concepte și algoritmi, ci să îi înțeleagă, să-i aplice și să-i adapteze la situații reale. Aici intervine valoarea metodei problematizării:

- permite dezvoltarea gândirii algoritmice și logice;
- încurajează creativitatea în soluționarea sarcinilor;
- ajută elevii să se obișnuiască cu procesul de analiză și testare;
- reproduce modul de lucru real în domeniul IT – bazat pe încercări, erori și rafinări continue.

Pentru a înțelege mai profund eficiența metodei problematizării în educație, este important să analizăm fundamentele teoretice care o susțin.

## 1.3 Ce spune teoria educațională?

Metoda problematizării se sprijină pe:

- Constructivism – elevul învață construindu-și cunoștințele prin explorare (Piaget, Bruner);
- Învățarea prin descoperire – cunoașterea este mai durabilă când este activ obținută, nu doar transmisă;
- Pedagogia competențelor – accentul se pune pe ce poate face elevul cu ceea ce învață;
- Problem-Based Learning (PBL) – foarte utilizat în educația STEM, presupune rezolvarea de situații reale.

Aceste principii teoretice se concretizează în practică printr-un mod specific de predare și învățare, care transformă atât rolul profesorului, cât și al elevului.

#### **1.4 Ce înseamnă să predai prin problematizare?**

Predarea prin problematizare reprezintă o strategie didactică esențială pentru dezvoltarea competențelor profesionale specifice domeniului studiat, întrucât stimulează gândirea critică, implicarea activă și învățarea reflexivă atât pentru profesori, cât și pentru elevi.

Pentru profesor, a predă prin problematizare înseamnă:

- să formuleze întrebări deschise și stimulatoare;
- să provoace gândirea elevilor, nu doar să ofere răspunsuri;
- să creeze contexte de învățare autentice, relevante;
- să accepte că uneori răspunsul vine din dezbateri, nu din manual.

Pentru elev, înseamnă:

- implicare activă în procesul de învățare;
- colaborare și argumentare cu colegii;
- învățare prin efort propriu și reflecție.

Prin urmare, pentru a valorifica pe deplin potențialul metodei problematizării în dezvoltarea competențelor profesionale specifice domeniului studiat, este necesară o implementare structurată și adaptabilă, prezentată în etapele ce urmează.

## **II. Etapele de implementare**

Această secțiune oferă o structură clară și etapizată pentru integrarea metodei problematizării în procesul educațional. Etapele propuse pot fi adaptate ușor în funcție de specificul clasei, nivelul elevilor și competențele prevăzute în Curriculum.

### **2.1 Planificarea didactică și integrarea metodei**

Implementarea eficientă a metodei problematizării începe cu o planificare didactică atentă, care presupune integrarea strategică a acestei abordări în proiectarea curriculară, în funcție de tematicile abordate, obiectivele de învățare și particularitățile elevilor.

- Identificarea unităților tematice în care se pretează aplicarea metodei problematizării (ex. „Programarea structurată”, „Informatica”, „Programarea procedurală”).
- Stabilirea obiectivelor centrate pe competențe și a rezultatelor așteptate.
- Includerea întrebărilor problematizante în secțiunile de predare din proiectarea anuală și semestrială.
- Adaptarea conținuturilor în funcție de nivelul clasei și ritmul de învățare.

Sugestii pentru debutanți: *Începe cu una sau două lecții-cheie, în care să introduci câte o întrebare provocatoare și să observi cum reacționează elevii. Ajustează pe parcurs.*

| Modulul                 | Unitatea de conținut                | Întrebare – cheie                                                                                                                                                                                                                                                                                                       | Impact                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Programarea structurată | Instrucțiunea repetitivă <b>for</b> | <p>Cum putem parcurge eficient o mulțime de date atunci când știm exact de câte ori trebuie să repetăm o acțiune?</p> <p>Este posibil să generăm toate numerele pare de la 1 la 100 fără să folosim o structură repetitivă? Dacă nu, cum ne ajută structura <b>for</b> să facem acest lucru mai simplu și eficient?</p> | <p>Această întrebare invită elevii să descopere <i>de ce</i> și <i>când</i> folosim instrucțiunea <b>for</b>, stimulând gândirea critică despre controlul execuției în programare și avantajele față de alte tipuri de bucle (cum ar fi <b>while</b>).</p> <p>Această întrebare provoacă elevii să analizeze utilitatea buclei <b>for</b> în generarea de secvențe controlate, mai ales când există un pas constant.</p> |
| Programarea procedurală | Fișerele                            | <p>Cum putem păstra datele introduse de utilizator chiar și după ce programul s-a încheiat?</p> <p>Ce se întâmplă cu datele unui program dacă acesta lucrează cu sute de valori și nu folosim fișiere pentru a le salva?</p>                                                                                            | <p>Această întrebare stimulează curiozitatea despre persistența datelor și conduce natural spre explorarea conceptului de fișiere ca soluție pentru memorarea datelor în afara memoriei volatile. Această întrebare provoacă elevii să reflecteze asupra limitărilor memoriei RAM și nevoii de stocare externă, deschizând discuția despre importanța fișierelor în programele reale.</p>                                |

Odată conturate unitățile și întrebările-cheie prin care se aplică metoda problematizării, următorul pas firesc este dezvoltarea unor resurse educaționale adecvate, care să susțină acest demers activ și exploratoriu.

## 2.2 Crearea resurselor educaționale

Pentru a susține aplicarea metodei problematizării în activitatea didactică, este esențială crearea unor resurse educaționale variate și relevante, care să stimuleze reflecția, analiza critică și rezolvarea creativă a situațiilor de învățare.

- Elaborarea de fișe de lucru bazate pe probleme reale sau simulate.
- Utilizarea planșelor didactice aprobate (cu suport vizual și explicativ).
- Integrarea activităților interactive în platforma Moodle (quiz-uri, forumuri, simulări).
- Adaptarea materialelor existente la formatul „problemă – analiză – soluție”.

**Situație-problemă:** Ești stagiar într-un departament de analiză statistică. Ai primit un set de date reprezentând temperaturile zilnice înregistrate într-un oraș timp de 30 de zile, stocate într-un vector liniar.

Superiorul tău dorește să afli: media temperaturilor, cea mai mare și cea mai mică valoare, zilele în care temperatura a fost egală cu media.

Sarcini: Descrie modul în care ai stoca și procesa datele.

- Ce tip de variabilă folosești?
- Cum parcurgi vectorul pentru a calcula media?

Cum identifici valorile minime și maxime?

- Ce strategie aplici în algoritm?

Cum determini pozițiile (zilele) în care temperatura a fost egală cu media?

- Ce verificare faci pentru fiecare element din vector?

Ce se poate întâmpla dacă nu inițializezi corect vectorul sau variabilele?

- Dă un exemplu de rezultat greșit și cum îl eviți.

Resursele create devin eficiente doar dacă sunt integrate într-un cadru interactiv, în care elevii sunt provocați să gândească, să colaboreze și să caute singuri soluții. Astfel, ajungem la modul concret în care se desfășoară lecțiile prin problematizare.

### 2.3 Desfășurarea lecțiilor interactive

Desfășurarea lecțiilor interactive constituie momentul-cheie al aplicării metodei problematizării, prin care profesorul creează un mediu stimulat de explorare, colaborare și construire activă a cunoașterii împreună cu elevii.

- Lansarea lecției prin întrebări deschise sau situații-problemă.
- Organizarea elevilor în grupe mici pentru brainstorming și discuții.
- Ghidarea rezolvării fără a oferi imediat răspunsul corect.
- Facilitarea schimbului de idei și încurajarea formulării de ipoteze multiple.

Rolul profesorului: *Este de facilitator, nu de evaluator. Ascultă, întreabă, provoacă elevul să-și susțină logica.*

Pentru ca metoda problematizării să își atingă întregul potențial formativ, este esențial ca și evaluarea să reflecte acest tip de învățare activă. În locul unei simple testări a cunoștințelor, accentul cade pe evaluarea procesului de gândire, a argumentării și a capacității de rezolvare a problemelor.

### 2.4 Evaluarea învățării prin problematizare

Evaluarea învățării prin problematizare presupune o abordare centrată pe proces și progres, valorificând metode formative care evidențiază modul în care elevii gândesc, argumentează și aplică cunoștințele în contexte autentice.

Utilizarea evaluării formative, prin:

- rubrici de evaluare (cu criterii pentru gândire logică, soluționare, argumentare);
- autoevaluare și reflecție individuală;
- proiecte de grup bazate pe probleme reale.
- acordarea feedback-ului orientat spre progres, nu doar notare.

| <i><b>Criteriu de evaluare</b></i> | <i><b>Punctaj (3p.)</b></i>   | <i><b>Punctaj (2p.)</b></i>    | <i><b>Punctaj (1p.)</b></i> |
|------------------------------------|-------------------------------|--------------------------------|-----------------------------|
| Înțelegerea problemei              | Clar formulată și analizată   | Parțial clară                  | Ambiguă sau lipsă           |
| Logica soluției                    | Coerentă, argumentată         | Cu unele ezitări               | Incompletă                  |
| Implementarea                      | Corectă, eficientă            | Funcțională, dar suboptimizată | Greșită sau lipsă           |
| Reflecția                          | Argument logic despre soluție | Reflecție sumară               | Nicio reflecție             |

Evaluarea nu încheie procesul, ci oferă puncte de reflecție pentru îmbunătățire. Prin urmare, este important ca profesorul să monitorizeze constant impactul metodei, să culeagă feedback și să ajusteze strategiile în funcție de nevoile reale ale elevilor.

## **2.5 Monitorizarea și adaptarea continuă**

Monitorizarea și adaptarea continuă sunt esențiale pentru asigurarea eficienței metodei problematizării, permițând ajustarea permanentă a strategiilor didactice pe baza observațiilor, feedback-ului elevilor și colaborării între cadrele didactice.

- Colectarea de feedback de la elevi (verbal sau prin chestionare).
- Observarea gradului de implicare și progresul în autonomie.
- Ajustarea lecțiilor și resurselor în funcție de rezultate.
- Discuții în cadrul catedrei pentru schimb de bune practici.

În instituțiile în care metoda este validată și aprobată (precum CEITI), se recomandă includerea acestuia în analiza periodică a activității didactice și în proiectările anuale.

Exemple de întrebări de feedback (pentru elevi):

- Ce parte a lecției te-a pus cel mai mult pe gânduri?
- Care întrebare ți-a schimbat modul de a vedea problema?
- Ce ai face diferit dacă ai rezolva aceeași sarcină a doua oară?

Astfel, odată ce procesul de monitorizare și adaptare confirmă eficiența metodei problematizării, devine evident că succesul acesteia depinde în mare măsură de rolul activ și reflexiv al profesorului în cadrul lecției.

## **III. Rolul profesorului în predarea prin problematizare**

Într-o lecție construită pe metoda problematizării, rolul profesorului se schimbă fundamental: de la transmitător de informații, la facilitator al gândirii, ghid al reflecției și partener în căutarea de soluții. Pentru un profesor debutant, această schimbare de perspectivă este esențială și adesea provocatoare, dar profund formativă.

### 3.1 Profesorul ca facilitator

Unul dintre cele mai importante roluri pe care le asumă profesorul în predarea prin problematizare este cel de facilitator al învățării, sprijinind dezvoltarea gândirii autonome și a spiritului critic prin întrebări provocatoare și contexte de explorare activă.

Facilitatorul este cel care:

- provoacă elevii să gândească, nu le oferă direct răspunsurile;
- creează contexte de învățare stimulante și relevante;
- susține procesul de descoperire, chiar și atunci când acesta implică greșeli sau incertitudine.

Exemplu de intervenție a profesorului facilitator:

- Interesantă soluția ta. Ce s-ar întâmpla dacă înlocuim această condiție cu o altă variabilă?
- Cum ai convinge un coleg că metoda ta este mai eficientă?

Rolul de facilitator nu se limitează la intervenții punctuale în lecție. Pentru ca elevii să răspundă la provocări și să se implice activ, este esențial ca profesorul să creeze un climat în care elevii se simt în siguranță să întrebe, să greșească și să exploreze idei fără teama de judecată.

### 3.2 Construirea unui climat de învățare deschis

Pentru a sprijini dezvoltarea gândirii critice și a unei înțelegeri profunde, este esențială construirea unui climat de învățare deschis, în care elevii se simt încurajați să exploreze, să greșească și să învețe împreună.

Un climat de învățare care sprijină problematizarea:

- valorizează întrebările și curiozitatea elevilor;
- încurajează exprimarea ipotezelor, chiar și a celor greșite;
- normalizează eroarea ca parte din procesul de învățare;
- promovează colaborarea și dialogul argumentativ.

Sugestii pentru debutanți:

- Creează „colțul întrebărilor” în clasă sau în platforma Moodle;
- Propune o regulă simplă: „*Greșelile ne ajută să gândim*”;
- Încurajează răspunsuri multiple la o singură problemă.

Odată construit un mediu prietenos și deschis, următorul pas este ca profesorul să stăpânească arta formulării întrebărilor provocatoare. Acestea sunt cheia declanșării procesului de problematizare și pot transforma o lecție obișnuită într-o experiență de învățare profundă.

### 3.3 Ghid pentru formularea întrebărilor provocatoare

Formularea întrebărilor provocatoare reprezintă o abilitate esențială pentru cadrele didactice, deoarece întrebările bine alese pot ghida elevii spre descoperire, reflecție și învățare profundă. În contextul informaticii, în loc de a întreba „*Ce este o variabilă?*”, o întrebare provocatoare ar putea fi: „*Cum ai convinge un coleg că alegerea tipului de variabilă poate influența performanța unui program?*”. O astfel de întrebare deschide discuții despre eficiență, optimizare și luarea deciziilor în programare.

Întrebările sunt nucleul metodei problematizării. O întrebare bună:

- nu are un singur răspuns corect;

- stimulează analiza, comparația, decizia;
- deschide posibilități de aplicare în viața reală.

Exemple de întrebări provocatoare în informatică:

- Cum ai optimiza un algoritm care funcționează corect, dar lent?
- Ce ai face dacă baza de date returnează rezultate greșite în funcție de filtru?
- Care este riscul dacă ignori validarea datelor într-o aplicație?
- Cum ai proiecta un sistem de înregistrare a utilizatorilor care să fie atât eficient, cât și sigur?

Cadru de formulare (model simplu pentru debutanți):

- *Ce s-ar întâmpla dacă...*
- *Cum ai aborda problema în cazul în care...*
- *Care ar fi cea mai eficientă soluție pentru...*

## IV. Resurse utile pentru aplicare

Integrarea metodei problematizării în predarea informaticii este susținută de un set de resurse create, testate și validate pe parcursul mai multor ani de predare și formare. Acestea sunt gândite să faciliteze munca profesorului și să sprijine elevul în procesul de învățare activă, prin sarcini reale și provocatoare.

### 4.1 Fișe metodologice (RED)

Au fost elaborate fișe metodologice sub formă de Resurse Educaționale Deschise (RED), aprobate și publicate online, accesibile gratuit pentru toți profesorii. Ele includ:

- situații-problemă adaptate disciplinelor TIC;
- sarcini gradate pe niveluri de dificultate;
- propuneri de reflecție și autoevaluare.

Utilizare practică: Profesorul poate prelua fișa și o poate adapta pentru grupuri de elevi cu ritmuri diferite de învățare sau poate folosi întrebările ca punct de plecare pentru o lecție întreagă.

Pe lângă fișele RED, care oferă flexibilitate și accesibilitate, un sprijin important în aplicarea metodei problematizării îl oferă și planșele didactice. Acestea aduc un plus de claritate prin reprezentări vizuale adaptate specificului informaticii.

### 4.2 Setul de planșe didactice aprobate

Disponibil în format tipărit și digital (cu ISBN), setul de planșe didactice pentru informatică este conceput pentru a sprijini:

- explicarea vizuală a noțiunilor algoritmice;
- facilitarea înțelegerii structurii codului;
- învățarea prin asocieri vizuale și simbolice.

Completarea materialelor vizuale cu activități structurate este esențială pentru o predare coerentă prin problematizare. De aceea, broșura oferă și fișe-model gata de utilizare, menite să sprijine aplicarea concretă a acestei metode în clasă.

### 4.3 Exemple de sarcini și activități

Broșura include un set de fișe-model (vezi Anexa II), care pot fi preluate și adaptate ușor. Acestea respectă structura:

- Context real / situație simulată;
- Întrebare provocatoare;
- Sarcini individuale și de grup;
- Spațiu pentru reflecție și feedback.

Sugestie pentru debutanți: *Folosește o singură fișă la începutul unui modul nou. Observă reacția elevilor și ajustează nivelul de provocare.*

Într-un demers de digitalizare și extindere a accesului la învățare interactivă, metoda problematizării a fost integrată și într-un curs online pe platforma Moodle, care valorifică tehnologia pentru a încuraja explorarea, colaborarea și gândirea critică.

## V. Recomandări pentru profesorii debutanti

Această secțiune oferă sugestii simple, aplicabile imediat în sala de clasă, menite să sprijine profesorii debutanți în primele încercări de a preda informatica prin metoda problematizării. Accentul este pus pe învățare treptată, reflecție și încredere în proces.

### 5.1 Sfaturi practice pentru activități didactice

Aplicarea metodei problematizării nu presupune schimbări radicale de la început; cu pași mici și bine gândiți, poți integra această abordare în lecțiile tale. Iată câteva sfaturi practice care te pot ajuta să începi eficient:

- Începe cu o singură întrebare problematizantă în prima parte a lecției. Nu e nevoie să restructurezi totul de la început.
- Alege teme familiare elevilor – astfel le va fi mai ușor să se implice și să propună soluții.
- Acceptă răspunsuri multiple. Evită „răspunsul corect din prima” – problematizarea înseamnă explorare.
- Construiește lecția în jurul unei situații reale – mai bine un caz scurt și provocator decât o problemă lungă, abstractă.
- Alternează problematizarea cu explicația – când vezi că elevii se blochează, oferă un punct de sprijin și pornește din nou.

Chiar dacă aceste sugestii practice sunt utile la început, e firesc ca apariția unor dificultăți să ridice întrebări. Pentru a evita descurajarea, este important să cunoaștem care sunt cele mai frecvente erori în aplicarea metodei problematizării și cum pot fi prevenite.

## 5.2 Erori frecvente și cum le evităm

Pedagogia centrată pe problematizare implică și o învățare continuă din practică, iar recunoașterea unor erori frecvente ne poate ajuta să le evităm și să îmbunătățim procesul didactic.

? Întrebări prea vagi sau prea complicate: reformulează întrebarea astfel încât să includă un context clar și o direcție de lucru.

? Așteptarea unui singur răspuns „corect”: deschide discuția către mai multe ipoteze. Creează un spațiu sigur pentru greșeli.

? Lipsa timpului pentru reflecție: alocă 5–10 minute pentru ca elevii să își noteze ideile înainte de discuția de grup.

? Prea multă intervenție din partea profesorului: lasă elevii să greșească și să se corecteze reciproc – este parte din învățare.

Odată conștientizate aceste capcane, următorul pas este să învățăm cum să adaptăm metoda problematizării în funcție de specificul fiecărei clase, astfel încât să rămână eficientă și motivantă, indiferent de nivelul elevilor.

## 5.3 Când și cum adaptăm metoda în funcție de stilul de învățare a elevilor

Pentru ca metoda problematizării să fie eficientă, este important să o adaptăm în funcție de stilurile de învățare și nivelul de pregătire al elevilor, astfel încât toți să se poată implica activ și cu sens.

Dacă ai o clasă/grupă de elevi cu ritm mai lent:

- redu complexitatea problemei, dar menține întrebarea deschisă;
- lucrează pe grupe mici, cu sarcini diferențiate.

Dacă ai o clasă avansată:

- formulează provocări care implică luarea de decizii;
- solicită soluții optimizate și justificări.

În grupe mixte:

- creează „roluri” în grupuri – cercetător, evaluator, sintetizator;
- astfel, fiecare elev contribuie diferit, dar esențial.

Recomandare generală: *Tratează problematizarea ca pe un drum, nu ca pe o rețetă. Fiecare lecție devine un mic laborator de gândire, iar tu ești ghidul care aprinde curiozitatea și susține procesul de descoperire.*

## Concluzii

*„Educația nu înseamnă să umpli un vas, ci să aprinzi un foc.”*

(Plutarh)

Metoda problematizării este mai mult decât o strategie didactică: este o schimbare de perspectivă asupra învățării. Ea mută centrul de greutate de la livrarea informației către provocarea gândirii, de la predare către descoperire, de la pasivitate la implicare activă.

Integrarea metodei problematizării în predarea informaticii nu este un obiectiv în sine, ci un mijloc puternic pentru a dezvolta gândirea critică, autonomia și capacitatea de a lua decizii algoritmice. Prin întrebări bine alese, situații relevante și un climat deschis, profesorul nu doar predă conținut, ci formează minți flexibile și curioase.

Indiferent de nivelul clasei sau de experiența didactică, metoda poate fi adaptată treptat, începând cu un exercițiu sau o întrebare. Important este să rămânem conectați la nevoile reale ale elevilor și să cultivăm spiritul exploratoriu al informaticii.

Pentru profesorii debutanți, integrarea metodei problematizării poate părea inițial provocatoare. Însă fiecare întrebare bine formulată, fiecare elev care argumentează o soluție și fiecare lecție în care se construiește cunoașterea împreună reprezintă un pas spre un învățământ mai viu, mai eficient, mai relevant.

Această broșură își propune să fie un sprijin real, o sursă de idei și o invitație la reflecție. Nu oferă soluții definitive, ci deschide drumuri – la fel cum o face orice întrebare problematizantă bine gândită.

### Propuneri și recomandări:

- Exersează arta întrebărilor: investește timp în formularea de întrebări deschise, relevante și provocatoare. Întrebările bine gândite pot transforma o lecție obișnuită într-o experiență de învățare memorabilă.
- Ascultă activ și valorizează ideile elevilor: creează un spațiu în care fiecare răspuns este o oportunitate de învățare. Evită să corectezi imediat – încurajează argumentarea și autocorectarea.
- Adaptează conținutul la realitatea elevilor: leagă noțiunile de informatică de situații reale, aplicații din viața de zi cu zi sau din domeniul IT. Astfel, crești relevanța și motivația.
- Folosește greșeala ca resursă didactică: nu penaliza greșelile, ci explorează-le împreună cu elevii. Întreabă: „De ce crezi că această soluție nu funcționează?” sau „Cum am putea corecta?”
- Încurajează colaborarea și munca în echipă: problemele complexe se înțeleg mai ușor prin dialog. Creează contexte în care elevii lucrează împreună, își asumă roluri și învață unii de la alții.
- Reflectează constant la propria practică: după fiecare lecție, întreabă-te ce a funcționat, ce ai schimba și cum ai putea face problematizarea mai eficientă data viitoare.
- Fii deschis la experimentare: nu aștepta „lecția perfectă” pentru a încerca această metodă. Începe cu pași mici și ajustează pe parcurs. Fiecare întrebare testată este un progres.
- Continuă formarea profesională: caută resurse, participă la cursuri sau comunități de practică didactică. Învățarea continuă este cheia excelenței în predare.

## **Referințe bibliografice**

1. Cartaleanu, T. și al., Formarea de competențe prin strategii didactice interactive, Chișinău, Pro Didactica, 2008.
2. Cucoș C. Pedagogie. Ediția a III-a. Editura Polirom, 2014. 536 p. ISBN 9789734640416.
3. Pânișoara I-O. Profesorul de succes. Editura Polirom, 2015. 248 P. ISBN 9789734652778.
4. Masalagiu C., Asiminoaei I., Țibu M. Didactica predării informaticii. Editura Polirom, 2016. 312p. ISBN 9789734633883.

**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**  
**Centrul de Excelență în Informatică și Tehnologii Informaționale**  
**Catedra Informatică II**

**Anexa 1**

Coordonat Metodist:

\_\_\_\_\_ /Jumbei O./

APROBAT

Director adjunct în instruire și educație

\_\_\_\_\_ /Obadă L. /

**PROIECT DIDACTIC DE LUNGĂ DURATĂ**  
**LA INFORMATICĂ**

**Grupa:** \_\_\_\_\_

**Specialitatea:** \_\_\_\_\_

**Profesor:** \_\_\_\_\_

Numărul de ore conform Planului de învățământ: \_\_\_\_\_

Examinat la ședința catedrei de Informatică II

\_\_\_\_\_ Proces verbal Nr. 1  
Șef catedră \_\_\_\_\_ /Gribineț L. /

CHIȘINĂU

**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Centrul de Excelență în Informatică și Tehnologii Informaționale**

Coordonat Metodist:

\_\_\_\_\_ /Jumbei O./

APROBAT

Director adjunct în instruire și educație

\_\_\_\_\_ /Obadă L. /

**PROIECT DIDACTIC DE LUNGĂ DURATĂ LA MODULUL**

**F.01.O.010 Programarea structurată**

**Grupa:** \_\_\_\_\_

**Specialitatea:** \_\_\_\_\_

**Profesor:** \_\_\_\_\_

Numărul de ore conform Planului de învățământ:

Teorie \_\_\_\_\_ ore, laborator \_\_\_\_\_ ore, studiul individual ghidat de profesor \_\_\_\_\_ ore

Examinat la ședința catedrei de Informatică II

\_\_\_\_\_ Proces verbal Nr. 1  
Șef catedră \_\_\_\_\_ /Gribineț L. /

CHIȘINĂU

## Fișe cu sarcini diferențiate, pe nivele



### Tema: Elemente fundamentale ale limbajelor de programare

#### Nivel de bază: Inițiere și recunoaștere a unităților lexicale

##### Fișa 1: Vocabularul limbajului

- 1.1. *Problemă generativă:* Cum înțelege calculatorul cuvintele și simbolurile dintr-un program?
- 1.2. Examinează codul prezentat formulează răspunsuri la întrebările aferente:

```
#include <iostream>
using namespace std;
int main(){
    int suma = 10+5;
    cout << suma;
    cin >> a >> b;
    return 0;
}
```

- Subliniază în cod:
  - cuvintele cheie
  - identificatorii
  - simbolurile speciale (=, +, ;)
- Ce se întâmplă dacă scriem `int 123suma = 10;`? De ce?



### Tema: Elemente fundamentale ale limbajelor de programare

#### Nivel de bază: Inițiere și recunoaștere a unităților lexicale

##### Fișa 2: Tipuri de date

- 2.1. *Problemă generativă:* Cum spunem calculatorului ce fel de informație păstrăm?
- 2.2. *Activitate:* Specifică pentru fiecare valoare tipul de date asociat din C++:

- 'A' -> \_\_\_\_\_
- 3.14 -> \_\_\_\_\_
- 42 -> \_\_\_\_\_

- 2.3. *Provocare:* Scrie o declarație corectă pentru o variabilă care păstrează:

- o literă \_\_\_\_\_
- o temperatură în grade (reală) \_\_\_\_\_



## Tema: Elemente fundamentale ale limbajelor de programare

### Nivel mediu: Structuri de bază și logică matematică în programare

#### Fișa 3: Operatori și precedență

3.1. *Problemă generativă:* Cum știe C++ ordinea în care să calculeze expresiile?

3.2. Examinează codul prezentat formulează răspunsuri la întrebările aferente:

```
int x = 4+2*3;
```

- Ce valoare va avea x?
- Ce rezultat vom obține dacă folosim paranteze:  $(4 + 2) * 3$ ?
- Scrie o expresie care folosește +, \*, și - și are ca rezultat 10.



## Tema: Elemente fundamentale ale limbajelor de programare

### Nivel mediu: Structuri de bază și logică matematică în programare

#### Fișa 4: Variabile și constante

4.1. *Problemă generativă:* Când și de ce am dori să nu mai schimbăm o valoare?

4.2. Examinează codul prezentat formulează răspunsuri la întrebările aferente:

```
const float pi= 3.14;  
int raza = 5;  
float arie = pi * raza * raza;
```

- De ce pi este constantă?
- Ce eroare vom primi dacă scriem  $pi = 3.1415$ ?





## Tema: Elemente fundamentale ale limbajelor de programare

### Nivel avansat: Scrierea completă și corectă a unui program

#### Fișa 5: Structura unui program

- 5.1. *Problemă generativă:* Ce se întâmplă dacă uităm o parte din scheletul programului?
- 5.2. Examinează codul prezentat, identifică și corectează greșelile, apoi formulează răspunsuri la întrebările aferente:

```
int a, b;  
cout << "Introdu doua numere";  
cin >> a >> b;  
sum = a + b;
```

- Ce lipsește pentru ca programul să ruleze corect?
- Cum se declară și inițializează variabila **sum**?



## Tema: Elemente fundamentale ale limbajelor de programare

### Nivel avansat: Scrierea completă și corectă a unui program

#### Fișa 6: Conversii și tipuri de date

- 6.1. *Problemă generativă:* Cum combinăm întregi cu numere reale fără pierderi de date?
- 6.2. Examinează codul prezentat formulează răspunsuri la întrebările aferente:

```
int i = 5;  
float f = i / 2;
```

- Ce valoare va avea **f**?
- Cum modificăm expresia ca să obținem rezultat real (ex: 2.5)?
- Ce face conversia  $(\text{float})i / 2$ ?



**Un set de întrebări cheie, de la simplu la complex,  
pentru a ilustra necesitatea și utilitatea conceptelor**

**Tema: Liste simplu înlănțuite**

*Acest set de întrebări ghidează elevii de la înțelegerea limitărilor structurilor statice la aprecierea flexibilității și utilității structurilor dinamice de date, concentrându-se pe cazul specific al listelor simplu înlănțuite. Prin răspunsurile la aceste întrebări, ei vor putea demonstra necesitatea și avantajele utilizării acestui tip de structură de date.*

**Limitările structurilor statice**

1. Imaginează-ți că ai un număr fix de locuri pentru a stoca numele prietenilor tăi. Ce se întâmplă dacă îți faci prieteni noi și nu mai ai loc?
2. Dacă vrei să inserezi un prieten nou în mijlocul listei tale de nume, ce trebuie să faci cu ceilalți prieteni pentru a face loc? Este aceasta o operațiune simplă și rapidă?
3. Dacă un prieten se mută și vrei să-l ștergi din listă, ce se întâmplă cu spațiul rămas liber? Poți să-l folosești eficient pentru un alt prieten?
4. Dacă nu știi de la început câți prieteni vei avea, cât de mare ar trebui să faci lista inițială pentru a fi sigur că ai suficient spațiu? Care sunt dezavantajele dacă faci o listă prea mare?

**Ideea de legătură și flexibilitate**

5. În loc să stocăm prietenii unul lângă altul într-un spațiu fix, ce-ar fi dacă fiecare prieten ar avea o "etichetă" care ne spune unde se află următorul prieten din listă? Cum ne-ar ajuta această "etichetă" să adăugăm un prieten nou oriunde în listă?
6. Dacă folosim aceste "etichete", mai avem nevoie ca prietenii să fie stocați în locuri consecutive? Cum ar putea influența acest lucru modul în care folosim memoria calculatorului?
7. Cum am putea să ștergem un prieten din listă folosind aceste "etichete", fără a muta toți ceilalți prieteni?

**Construirea listei înlănțuite**

8. Cum am putea reprezenta fiecare "prieten" și "eticheta" sa într-o structură de date? Ce informații ar trebui să conțină fiecare element al acestei structuri? (Introducerea conceptului de nod cu date și pointer către următorul nod).
9. Cum am putea identifica începutul listei noastre de prieteni dacă aceștia nu sunt stocați în locuri consecutive? (Necesitatea unui pointer către primul nod - "capul" listei).
10. Cum am putea ști când am ajuns la sfârșitul listei noastre de prieteni folosind aceste "etichete"? (Necesitatea unui pointer nul la ultimul nod).

**Analiză comparativă și eficiență**

11. Compară operațiunea de inserare a unui element nou într-un array față de inserarea într-o listă simplu înlănțuită. În ce situații este mai eficientă lista înlănțuită?
12. Compară operațiunea de ștergere a unui element dintr-un array față de ștergerea dintr-o listă simplu înlănțuită. În ce situații este mai eficientă lista înlănțuită?
13. Care este avantajul listelor simplu înlănțuite în ceea ce privește utilizarea memoriei în comparație cu array-urile atunci când nu știm dimensiunea exactă a datelor?
14. Există operațiuni care sunt mai dificil de realizat sau mai puțin eficiente într-o listă simplu înlănțuită în comparație cu un array? De exemplu, cum am accesa al treilea element al listei?

**Aplicații practice și extinderi**

15. Poți să te gândești la situații reale în care ar fi util să adaugi sau să ștergi elemente dintr-o colecție de date frecvent, fără a cunoaște dimensiunea exactă de la început? (Exemple: gestionarea cozii de așteptare, istoricul navigării web, implementarea unei stive).
16. Cum ar putea fi folosite listele simplu înlănțuite pentru a implementa alte structuri de date mai complexe? (stive, cozi).
17. În contextul gestionării memoriei unui calculator, cum ar putea fi folosite idei similare cu cele din listele înlănțuite pentru a alocă și elibera blocuri de memorie în mod dinamic?

## Set de fișe diferențiate

### Structura decizională „if” în C++

Obiectiv: Sprijinirea învățării diferențiate a structurii decizionale if în C++ prin utilizarea fișelor problematizante adaptate pe niveluri de competență: nivel de bază, nivel mediu și nivel avansat.

#### I. Nivel de bază – Descoperirea condiționalului

Rezultatul învățării: Înțelegerea ideii de „dacă... atunci...”

##### *Fișa 1: Ce se întâmplă dacă plouă?*

**Problemă generativă:** Cum poate un program să ia decizii în locul nostru?

```
if (ploua) {
    cout << "Ia umbrela!"; }
```

**Întrebări:** Ce înseamnă `if (ploua)`? Ce se întâmplă dacă `ploua = false`?

##### *Fișa 2: Cine are acces la joc?*

**Problemă generativă:** Cum putem lăsa doar unii elevi să joace?

```
int varsta;
cin >> varsta;
if (varsta >= 10) {
    cout << "Poti juca!"; }
```

**Întrebări:** Ce se afișează dacă introducem 9? Ce am putea adăuga ca să scrie „Prea mic!” în rest?

##### *Fișa 3: Dacă elevul e premiant...*

**Problemă generativă:** Cum putem recompensa performanța?

```
bool premiant = true;
if (premiat) {
    cout << "Felicitări!"; }
```

**Sarcină:** Afișează și „Succes!” dacă **NU** e premiant.

##### *Fișa 4: Par sau impar?*

**Problemă generativă:** Cum identificăm dacă un număr este par?

```
int n;
cin >> n;
if (n % 2 == 0) {
    cout << "Par"; }
```

**Provocare:** Adaugă o ramură pentru „Impar”.

##### *Fișa 5: Lumina se aprinde doar noaptea*

**Problemă generativă:** Cum reacționează un program la condiții externe?

```
bool esteNoapte = true;
if (esteNoapte) {
    cout << "Aprinde lumina"; }
```

**Întrebare:** Ce se întâmplă dacă `esteNoapte` este false?



## II. Nivel mediu– Decizii compuse și gândire logică

Rezultatul învățării: Fiecare fișă conține o **situație provocatoare**, o **întrebare generativă** și o **sarcină de codificare cu reflecție**.

### Fișa 1 – Par sau impar cu o întorsătură logică

**Situație-problemă:** Un profesor vrea să afle dacă un număr introdus de elev este par. Dacă este par, acesta primește un punct bonus. Dar dacă numărul este și divizibil cu 10, primește două puncte.

**Întrebare generativă:** Poate un număr fi divizibil cu 10 și să nu fie par? Justifică.

**Sarcină:** Scrie un program care citește un număr întreg. Afișează:

- „1 punct bonus” dacă numărul e par;
- „2 puncte bonus” dacă e divizibil cu 10;
- „0 puncte” altfel.

Explică de ce este importantă ordinea testelor în acest caz.



### Fișa 2 – Elevul premiant sau nu?

**Situație-problemă:** Un elev este considerat premiant dacă are media peste 8 și nu are nicio notă sub 7.

**Întrebare generativă:** Este suficientă o singură condiție logică pentru a decide dacă un elev e premiant?

**Sarcină:** Scrie un program care primește 3 note ale unui elev. Calculează media. Verifică dacă elevul îndeplinește ambele condiții pentru a fi premiant. Explică de ce este necesar să folosim operatorul && în acest caz.



### Fișa 3 – Acces la laboratorul de informatică

**Situație-problemă:** Un elev poate intra în laboratorul de informatică doar dacă are ecuson și a obținut nota minimă 6 la testul precedent.

**Întrebare generativă:** Care este diferența între condițiile compuse și condițiile imbricate (if în if)?

**Sarcină:** Scrie un program care primește două informații: dacă are ecuson (1 sau 0) și nota elevului. Verifică dacă poate intra în laborator. Testează și explică diferența între:

if (are\_ecuson && nota >= 6) ... și if (are\_ecuson) if (nota >= 6) ...



## III. Nivel avansat – Structuri decizionale avansate

Rezultatul învățării: fișele pentru nivel avansat centrate pe structuri decizionale avansate, optimizare a condițiilor, evitarea redundanței, scriere eficientă a expresiilor logice și gândire algoritmică clară și riguroasă.

### Fișa 1 – Validarea unei date calendaristice

**Situație-problemă:** Un program de evidență academică trebuie să valideze dacă o dată introdusă de utilizator este una calendaristică corectă, ținând cont și de anii bisecți.

**Întrebare generativă:** Cum se pot combina mai multe condiții pentru a verifica toate cazurile fără a duce la cod redundant?

**Sarcină:** Scrie un program care: primește o zi, o lună și un an. Verifică dacă data este validă, folosește decizii imbricate și expresii logice optimizate, analizează corect anul bisect.



### Fișa 2 – Determinarea poziției unui punct față de un dreptunghi

**Situație-problemă:** Într-un sistem de planificare grafică (ex. aplicație de orar), se dorește identificarea poziției unui punct (x, y) față de un dreptunghi.

**Întrebare generativă:** Cum putem folosi combinații de condiții logice pentru a verifica poziția unui punct într-un spațiu 2D?

**Sarcină:** Scrie un program care: primește coordonatele unui punct și ale colțurilor dreptunghiului, afișează dacă punctul se află: în interior, pe latură, în exterior. Utilizează expresii logice clare, fără duplicare inutilă.



### Fișa 3 – Clasificator de secvențe numerice

**Situație-problemă:** Se dorește un algoritm care clasifică o secvență de trei numere în una dintre următoarele categorii: „crescătoare strict”, „descrescătoare strict”, „egală”, „fără ordine clară”.

**Întrebare generativă:** Cum putem testa toate variantele fără a scrie `if` suprapuse ineficient?

**Sarcină:** Scrie un program care primește trei numere reale, determină și afișează tipul secvenței, evită dublarea expresiilor logice și optimizează deciziile.



### Fișa 5 – Sistem de acordare a burselor

**Situație-problemă:** Bursele se acordă pe baza mediei și a altor criterii astfel:

- Media  $\geq 9.50$  + activitate: bursă de merit
- Media  $\geq 8.50$  fără activitate: bursă de performanță
- Media  $\geq 7.00$  și venituri sub 2500 lei: bursă socială
- Altfel: fără bursă

**Întrebare generativă:** Cum poți organiza testele astfel încât să nu apară suprapuneri sau contradicții?

**Sarcină:** Scrie un program care primește media, statutul activității extracurriculare și venitul pe membru, afișează tipul bursei atribuite, scrie expresiile logice eficient și în ordine corectă.



### Fișa 6 – Sistem de validare a parolelor

**Situație-problemă:** Pentru a crea un cont într-o aplicație, parola introdusă trebuie să respecte anumite cerințe:

Lungime minimă de 8 caractere:

- Să conțină cel puțin o literă mare
- Să conțină cel puțin o cifră
- Să conțină cel puțin un caracter special (ex: `!@#$%^&*`)

**Întrebare generativă:** Cum poți structura testele de validare pentru a optimiza verificările și a preveni redundanțele?

**Sarcină:** Scrie un program care primește o parolă și validează dacă aceasta respectă toate criteriile de securitate. Afișează un mesaj corespunzător pentru fiecare criteriu care nu este îndeplinit.



## Set de fișe integrate avansate

### Tipuri de date fișiere în C++

Fiecare fișă pornește de la o situație contextualizată din viața reală, cu cerințe escaladate pentru nivelele de bază, mediu, avansat, dar adresând și competențe algoritmice avansate: gestionarea fișierelor, manipularea stringurilor, optimizări și organizare de date structurate.

#### *Fișa 1: Jurnalul de activitate fizică*

**Context real:** O aplicație de fitness salvează zilnic în fișierul `activitate.txt` informații în format: zi, tip\_activitate, durata\_minute, calorii\_consumate.

**Nivel de bază:** Explică ce tipuri de date sunt potrivite pentru fiecare câmp. Ce tip de date alegi pentru durata și calorii?

**Nivel mediu:** Scrie un program care citește datele din fișier și calculează totalul kaloriilor consumate în ultima săptămână.

**Nivel avansat:** Afișează activitatea cu cea mai mare medie de calorii arse/minut și scrie într-un nou fișier doar acele activități care au durată > 45 min.



#### *Fișa 2: Gestionarea unei baze de date de utilizatori*

**Context real:** Un site web stochează date despre utilizatori în fișierul `utilizatori.txt`: id, nume\_utilizator, varsta, email.

**Nivel de bază:** Propune tipuri de date pentru fiecare câmp. Care dintre ele pot fi de tip șir de caractere?

**Nivel mediu:** Scrie un program care citește toți utilizatorii și îi afișează pe cei cu vârsta > 18 ani.

**Nivel avansat:** Grupând utilizatorii după prima literă din nume, creează câte un fișier pentru fiecare grup (ex: grupa\_A.txt, grupa\_B.txt etc.) și scrie în fiecare fișier utilizatorii corespunzători.



#### *Fișa 3: Aplicație de buget personal*

**Context real:** Un tânăr își înregistrează cheltuielile zilnice într-un fișier `buget.txt`: zi, categorie, sumă.

**Nivel de bază:** Alege tipurile de date potrivite. Ce date poți reprezenta cu numere reale și care sunt stringuri?

**Nivel mediu:** Scrie un program care citește fișierul și calculează totalul cheltuielilor pe categorii (alimentare, transport, etc.).

**Nivel avansat:** Identifică luna (ex: extrage din „zi” informația lunii) cu cele mai multe cheltuieli și creează un fișier care conține doar datele acelei luni.



#### *Fișa 5: Arhivă digitală de lucrări*

**Context real:** O platformă educațională salvează lucrările elevilor în fișierul `lucrari.txt`:

nume\_elev, disciplina, nota, titlu\_lucrare

**Nivel de bază:** Ce tipuri de date sunt potrivite pentru fiecare câmp? Justifică alegerea tipului pentru „nota”.

**Nivel mediu:** Scrie un program care afișează toate lucrările cu notă mai mare de 8, împreună cu numele elevului.

**Nivel avansat:** Afișează elevul cu cea mai mare medie pe disciplină și scrie într-un fișier separat titlurile lucrărilor sale.



### Criterii generale de evaluare:

| Criteriu evaluat                                        | Descriere                                                                  | Punctaj maxim     |
|---------------------------------------------------------|----------------------------------------------------------------------------|-------------------|
| Înțelegerea cerinței                                    | Corectitudinea alegerii tipurilor de date, justificarea logică             | 10 puncte         |
| Corectitudinea sintaxei și folosirea structurii de bază | Declarare corectă variabile, structură de bază a programului               | 10 puncte         |
| Prelucrarea datelor din fișier                          | Citirea, parcurgerea și manipularea datelor din fișier                     | 20 puncte         |
| Calculul corect și complet                              | Algoritm corect pentru procesare (medii, selecții, sortări etc.)           | 25 puncte         |
| Răspunsuri avansate / extensii                          | Afișări selective, salvări în fișiere, filtrare, structurare optimă        | 15 puncte         |
| Claritate, stil și organizare                           | Denumiri semnificative, indentare, comentarii, modularizare (dacă e cazul) | 10 puncte         |
| Inițiativă, originalitate (opțional)                    | Soluții proprii, optimizări, cod suplimentar explicat                      | 10 puncte         |
| <b>Total</b>                                            |                                                                            | <b>100 puncte</b> |

### Barem de corelare:

#### Punctaj obținut Nota

|          |    |
|----------|----|
| 91 – 100 | 10 |
| 81 – 90  | 9  |
| 71 – 80  | 8  |
| 61 – 70  | 7  |
| 51 – 60  | 6  |
| 41 – 50  | 5  |
| < 40     | 4  |

### Sugestii de aplicare diferențiată pe niveluri:

- **Nivel 5–6:** Se evaluează mai ales **înțelegerea conceptuală**, justificarea alegerilor de tipuri de date, folosirea simplă a fișierelor – max. 50 puncte.
- **Nivel 7–8:** Se pune accent pe **structura logică a codului**, procesarea corectă a datelor din fișier – max. 70–80 puncte.
- **Nivel 9–10:** Se urmărește **implementarea completă, extensibilă, eficientă**, inclusiv scrierea în fișiere noi – max. 100 puncte

Data:

Grupe:

Profesorul:

Cabinetul:

Disciplina: *Programarea calculatorului*

Unitatea de conținut: *Grafuri. Aplicații.*

Tipul lecției: *de formare a deprinderilor practice*

Durata: *90 min*

Unitate de competență: *UC4. Utilizarea algoritmilor bazați pe metode de reprezentare a grafurilor în cadrul aplicațiilor de consolă*

Obiectivele lecției:

- De înțelegere
  - O1 – să numească cele 4 metode de reprezentare a grafurilor.
  - O2 – să deducă algoritmul de creare a matricei de adiacență prin lista muchiilor/lista vecinilor
  - O3 – să reproducă algoritmul de parcurgere a grafurilor
- De aplicare
  - O4 – să implementeze cunoștințele în determinarea algoritmului de prelucrare a grafurilor
  - O5 – să scrie secvență de cod conform sarcinii propuse

Strategii didactice:

Metode de învățare și procedee de instruire:

- Metode de comunicare orală: conversația, explicația, discuția dirijată, problematizarea;

Forme de organizare a activităților:

- Frontală;
- Individuală.

Metode de evaluare:

- Observarea și aprecierea verbală.

Materiale didactice:

- Calculator;
- Tabla interactivă;
- Cursul pe Moodle (<https://moodle1.ceiti.md/course/view.php?id=3265&section=1> )

Referințe bibliografice:

- L. CHIRIAC, M. BOSTAN. Elemente de teoria grafurilor. Chișinău, ÎS Tipografia Centrală, 2022 - [1]
- <https://www.pbinfo.ro/> - [2]

**Scenariul didactic al lecției**

| Obiective operaționale       | Tehnologia realizării și conținuturile vizate                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                | Strategii didactice                         |                                                                                                                                                                                                                                                                                                                                               |                       |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
|                              | Activitatea profesorului                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Activitatea elevilor                                           | Metode de învățare și procedee de instruire | Materiale didactice                                                                                                                                                                                                                                                                                                                           | Forme de organizare   |
| Evocarea – 25 min            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                |                                             |                                                                                                                                                                                                                                                                                                                                               |                       |
| O1                           | Provoacă elevii să determine cele 4 reprezentări a grafurilor: matricea de adiacență, matricea de incidență, lista muchiilor, lista vecinilor.<br><br>Rezolvă împreună cu elevii sarcina 412 de pe pbinfo.ro.                                                                                                                                                                                                                                                                                               | Participa activ la discutii și se implică în scrierea codului. | Discuție dirijată                           | Tabla interactiva<br><br><a href="https://www.pbinfo.ro/probleme/412/adiacenta">https://www.pbinfo.ro/probleme/412/adiacenta</a>                                                                                                                                                                                                              | Frontala              |
| Realizarea sensului – 60 min |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                |                                             |                                                                                                                                                                                                                                                                                                                                               |                       |
| O2                           | Explică <u>parcurgerea grafurilor neorientate</u> :<br>1. <u>Parcurgerea în adâncime</u> DFS(Depth-First Search) – explică algoritmul și scrierea codului<br>2. <u>Parcurgerea în lățime</u> BFS(Breadth-First Search) – explică algoritmul și scrierea codului.<br><br>Propune elevelor să exerseze la tablă un exemplu Anexa 1.<br><br>Propune elevilor să se conecteze pe site-ul specializat <a href="https://www.pbinfo.ro">https://www.pbinfo.ro</a> , unde este generat un laborator spre realizare. | Participa activ la discutii                                    | Discuție dirijată                           | <a href="https://www.cs.usfca.edu/~galles/visualization/DFS.html">https://www.cs.usfca.edu/~galles/visualization/DFS.html</a><br><br><a href="https://www.cs.usfca.edu/~galles/visualization/BFS.html">https://www.cs.usfca.edu/~galles/visualization/BFS.html</a><br><br><a href="https://graphonline.ru/en/">https://graphonline.ru/en/</a> | Frontala              |
| O3                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Trece la tabla                                                 | Problematizare                              | Tabla interactivă                                                                                                                                                                                                                                                                                                                             | Frontala              |
| O2                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Participa activ la discutii                                    |                                             | Laboratorul virtual                                                                                                                                                                                                                                                                                                                           | Frontala              |
| O3                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Realizează sarcinile                                           | Problematizare                              |                                                                                                                                                                                                                                                                                                                                               | Individual            |
| O4<br>O5                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                |                                             |                                                                                                                                                                                                                                                                                                                                               |                       |
| Reflectare -5 min            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                |                                             |                                                                                                                                                                                                                                                                                                                                               |                       |
| O5                           | Propune elevilor să finalizeze laboratorul în termeni cât mai restrânși.                                                                                                                                                                                                                                                                                                                                                                                                                                    | Participa activ la discutii                                    | Discuție dirijată                           | Chestionar Moodle                                                                                                                                                                                                                                                                                                                             | Frontal<br>Individual |

**Rubrică de evaluare – Învățare prin problematizare**

| <b>Criterii de evaluare</b>                         | <b>Excelent (4)</b>                                                                       | <b>Bine (3)</b>                                                           | <b>Satisfăcător (2)</b>                                                | <b>Începător (1)</b>                                                       |
|-----------------------------------------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------|----------------------------------------------------------------------------|
| Înțelegerea problemei                               | Identifică toate elementele esențiale ale problemei și relațiile dintre ele               | Identifică majoritatea elementelor relevante ale problemei                | Identifică doar elemente de bază ale situației problematice            | Are dificultăți în înțelegerea cerinței sau interpretează eronat problema  |
| Formularea ipotezelor și a strategiilor             | Propune ipoteze/logici clare, valide și strategii eficiente                               | Propune soluții fezabile și parțial optimizate                            | Propune strategii parțiale sau incomplet susținute de ipoteze          | Nu formulează ipoteze coerente sau folosește strategii inadecvate          |
| Aplicarea cunoștințelor și transferul în rezolvare  | Integrează corect cunoștințe anterioare, folosind concepte noi în mod creativ și eficient | Aplică cunoștințele în mod corect, dar cu unele ezitări în transfer       | Aplică parțial conceptele în situația nouă                             | Nu reușește să transfere cunoștințele în contextul problemei               |
| Rezolvare logică și structurare algoritmică         | Soluția este logică, clar structurată și eficientă (optimizată unde este cazul)           | Soluția este logică, dar cu mici inadvertențe în organizare sau eficiență | Soluția este funcțională dar greoaie, neoptimizată sau cu erori minore | Soluția este incoerentă sau conține erori conceptuale majore               |
| Justificarea deciziilor/<br>Reflecție metacognitivă | Argumentează clar alegerile făcute și reflectă asupra validității lor                     | Oferă justificări parțiale sau intuitive pentru deciziile sale            | Răspunde superficial, fără justificări clare                           | Nu oferă nicio justificare sau justificările sunt eronate / contradictorii |
| Autonomie și implicare în rezolvare                 | Demonstrează autonomie totală în procesul de rezolvare și implicare activă                | Lucrează autonom cu puțină îndrumare                                      | Are nevoie de sprijin în etape-cheie                                   | Nu poate progresa fără intervenția profesorului                            |

## Set de fișe integrate avansate

## Metoda backtracking în C++

Fiecare fișă este structurată după modelul: **situație-problemă**, **întrebare generativă**, **sarcină practică**, și sunt potrivite pentru evaluare prin rubrică.

**Fișa 1 – Generarea tuturor permutărilor**

**Situație-problemă:** Un profesor vrea să genereze toate variantele posibile de ordonare a unui grup de 3 elevi care participă la o activitate, dar fără repetiții.

**Întrebare generativă:** Cum poate un program „să se răzgândească” dacă a făcut o alegere greșită într-o secvență?

**Sarcină:**

- Scrie un program care generează toate permutările a  $n$  numere distincte.
- Utilizează metoda backtracking cu vector de marcaj.
- Afișează toate permutările în ordine lexicografică.

**Fișa 2 – Colorarea hărții (problema celor  $n$  culori)**

**Situație-problemă:** Trebuie să colorezi harta facultății astfel încât nicio sală alăturată să nu aibă aceeași culoare. Ai un număr limitat de culori.

**Întrebare generativă:** De ce metoda backtracking este preferabilă în cazul în care numărul de combinații crește exponențial?

**Sarcină:**

- Scrie un program care colorează  $n$  noduri (săli), respectând constrângerea că nodurile adiacente nu pot avea aceeași culoare.
- Folosește matricea de adiacență și metoda backtracking.
- Testează pentru 3 sau 4 săli și 2–3 culori.

**Fișa 3 – Submulțimi cu sumă fixă**

**Situație-problemă:** Dorești să găsești toate combinațiile de cursuri care însumează exact 20 de credite, având o listă cu valoarea în credite pentru fiecare curs.

**Întrebare generativă:** Cum alegi între explorarea în adâncime (backtracking) și o soluție iterativă?

**Sarcină:**

- Scrie un program care, dată o mulțime de  $n$  numere naturale și o sumă  $S$ , generează toate submulțimile care însumează exact  $S$ .
- Folosește backtracking pentru a include sau exclude fiecare element.
- Afișează toate combinațiile valide.

**Fișa 4 – Probleme de parantezare corectă**

**Situație-problemă:** Vrei să generezi toate combinațiile valide de paranteze care pot fi folosite într-o expresie C++ cu  $n$  perechi de paranteze.

**Întrebare generativă:** Cum poate backtrackingul să „taie ramurile” care duc sigur spre soluții incorecte?

**Sarcină:**

- Scrie un program care generează toate șirurile corecte de paranteze (ex: pentru  $n = 3 \rightarrow ((())), ((()())),$  etc.).
- Aplică metoda backtracking cu două contori: pentru deschideri și închideri.
- Nu permite ca numărul parantezelor închise să-l depășească pe cel deschis.



### **Fișa 5 – Labirintul facultății (problema drumului în matrice)**

**Situație-problemă:** Un student rătăcit vrea să ajungă la bibliotecă pornind dintr-un colț al hărții facultății, reprezentată ca o matrice cu obstacole.

**Întrebare generativă:** Cum controlezi „întoarcerea” dintr-o stare invalidă într-un spațiu bidimensional?

**Sarcină:** Scrie un program care găsește toate drumurile posibile de la poziția (0,0) la (n-1,n-1) într-o matrice de  $n \times n$ , evitând obstacole (celule marcate cu 1). Aplică metoda backtracking și marchează traseul în mod recursiv. Afișează drumurile posibile.

*PS: Aceste fișe stimulează: gândirea algoritmică recursivă; gestionarea stărilor parțiale; decizia de continuare sau retragere; optimizarea prin reducerea căutărilor inutile.*

**Barem de evaluare** adaptat pentru fișele problematizante pe tema **metodei backtracking** în C++, centrat pe competențe avansate precum:

- înțelegerea stărilor recursive,
- construcția condițiilor de continuare și revenire,
- optimizarea prin reducerea spațiului de căutare.

**BAREM DE NOTARE – Metoda Backtracking (Nivel avansat)**

| Criteriu de evaluare                                                          | Punctaj maxim    |
|-------------------------------------------------------------------------------|------------------|
| <b>1. Înțelegerea problemei și a conceptului de backtracking</b>              |                  |
| Identificarea cerinței algoritmice și formularea stărilor inițiale și finale  | 2 p              |
| Recunoașterea momentului de ieșire din recursie și condiției de <b>oprire</b> | 2 p              |
| <b>2. Construcția corectă a algoritmului recursive</b>                        |                  |
| Utilizarea structurii recursive corecte (apel recursiv, revenire, ramificare) | 2 p              |
| Verificarea condiției de continuare validă la fiecare pas                     | 2 p              |
| Utilizarea unui vector de marcaj / stări parțiale (unde e cazul)              | 1 p              |
| <b>3. Claritatea, corectitudinea și optimizarea soluției</b>                  |                  |
| Cod bine organizat, lizibil, fără cod redundant                               | 1 p              |
| Evitarea cazurilor redundante / ineficiente prin filtrare logică timpurie     | 1 p              |
| <b>4. Reflecție / Justificare logică (scrisă sau orală)</b>                   |                  |
| Explicarea structurii de control și a succesiunii apelurilor recursive        | 1 p              |
| Justificarea ordinii testelor și a strategiei de căutare (DFS, revenire etc.) | 1 p              |
| <b>TOTAL</b>                                                                  | <b>13 puncte</b> |

**Interpretare sumativă (opțională):**

| Punctaj total | Nivel de performanță  | Interpretare                                            |
|---------------|-----------------------|---------------------------------------------------------|
| 12–13         | Excelent              | Soluție completă, optimizată, clară                     |
| 10–11         | Foarte bine           | Soluție corectă cu îmbunătățiri minore                  |
| 8–9           | Bine                  | Funcțional, dar cu lipsuri în claritate sau eficiență   |
| 6–7           | Satisfăcător          | Funcțional parțial, dar cu erori sau confuzii logice    |
| Sub 6         | Necesită îmbunătățiri | Concepte greșit aplicate sau lipsa unei strategii clare |