

Colegiul Național „Grigore Moisil”  
Urziceni, Ialomița

**CURRICULUM DE INFORMATICĂ**  
**Disciplină opțională - CDȘ**

**LIMBAJE DE PROGRAMARE**

CLASA a IX-a REAL

Autor: Prof. Bardac Sorin

**ARGUMENT. NOTĂ DE PREZENTARE.**

Pornind de la beneficiile pe care le poate produce realizarea unei legături mult mai vizibile între învățarea școlară și aplicarea socială a rezultatelor acestei învățări, rezultă necesitatea accentuării laturii pragmatice a aplicării curriculum-ului, profesorul fiind cel ce face legătura directă și evidentă între **ce se învață** și **de ce se învață**.

Programa actuală trebuie înțeleasă ca o etapă necesară pentru crearea unei expertize adecvate și este deschisă revizuirilor ulterioare.

Noțiunile folosite sunt cele uzuale: *competențele generale* se referă la ansambluri structurate de cunoștințe și deprinderi, care se definesc pe obiect de studiu și se formează pe durata ciclului de învățământ. În cazul prezentei discipline opționale, ele sunt preluate de la disciplina de trunchi comun pe care o completează – *Informatica*. *Competențele specifice* se definesc pe obiect de studiu și se formează pe durata unui an de studiu, fiind deduse din competențele generale.

Având în vedere finalitatea aplicativă imediată a conținutului teoretic, orele se vor desfășura în laborator;

Ordinea în parcurgerea conținuturilor rămâne la latitudinea profesorului, acesta având libertatea de a introduce elemente noi, specifice;

Predarea va fi orientată pe *rezolvarea de probleme*, utilizându-se preponderent *metoda învățării prin descoperire* și punându-se accent pe *analiza problemei*.

Realizarea proiectelor în cadrul activităților practice va urmări dezvoltarea abilităților de lucru în echipă.

## COMPETENȚE GENERALE:

1. Elaborarea algoritmilor de rezolvare a problemelor
2. Implementarea algoritmilor într-un limbaj de programare
3. Aplicarea algoritmilor fundamentali în prelucrarea datelor

### Competența generală:

1. Elaborarea algoritmilor de rezolvare a problemelor

Competențe specifice	Conținuturi
<ul style="list-style-type: none"><li>• Analizarea enunțului unei probleme și stabilirea pașilor de rezolvare a problemei.</li><li>• Reprezentarea algoritmilor în limbaj natural, schemă logică, pseudocod, limbaje de programare.</li><li>• Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.</li></ul>	<p><b>Etapele rezolvării problemelor.</b> Rezolvări umane și rezolvări cu ajutorul calculatorului. Exemple.</p> <p><b>Noțiunea de algoritm.</b> Caracteristici. Exemple.</p> <p><b>Reprezentarea algoritmilor.</b> Limbaje pentru oameni: - Limbajul natural; - Schema logică; - Pseudocod. Limbaje pentru calculatoare. - Limbaje de nivel înalt; - Limbaj de asamblare; - Binar executabil. Trecerea de la o reprezentare la alta. Exemple.</p> <p><b>Principiile programării structurate.</b> Structuri de bază: - structura liniară, - structura alternativă, - structura repetitivă. Teorema Bohm – Jacopini (1966).</p> <p><b>Eficiența și complexitatea algoritmilor.</b></p>

**Competența generală:**

## 2. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice	Conținuturi
<ul style="list-style-type: none"> <li>• Transcrierea corectă a algoritmilor din pseudocod într-un limbaj de programare (limbajul Pascal, limbajul C/C++, limbajul Java etc.).</li> <li>• Utilizarea tipurilor de date potrivite specificului problemei.</li> <li>• Folosirea structurilor de programare oferite de limbaj în mod corect și adecvat problemei.</li> <li>• Utilizarea fișierelor text pentru citirea datelor și salvarea rezultatelor.</li> <li>• Utilizarea unui mediu de programare pentru limbajul studiat.</li> </ul>	<p><b>Elementele de bază ale limbajului de programare</b></p> <p><b>Noțiuni introductive</b></p> <ul style="list-style-type: none"> <li>• Structura programelor</li> <li>• Vocabularul limbajului</li> <li>• Tipuri simple de date (standard)</li> <li>• Constante, variabile, expresii</li> <li>• Citirea/scrierea datelor</li> <li>• Instrucțiunea de atribuire</li> <li>• Operații permise asupra diferitelor tipuri de date</li> <li>• Rezultatul fiecărei operații și tipul său de dată</li> </ul> <p><b>Structuri de control</b></p> <ul style="list-style-type: none"> <li>• Structura liniară</li> <li>• Structura alternativă</li> <li>• Structuri repetitive</li> <li>• Imbricarea structurilor de programare</li> </ul> <p><b>Fișiere text.</b> Declarare Operații specifice</p> <p><b>Mediul de programare al limbajului studiat</b></p> <ul style="list-style-type: none"> <li>• Prezentare generală</li> <li>• Editarea programelor sursă</li> <li>• Compilarea și rularea programelor</li> <li>• Depanarea programelor</li> </ul>

**Competența generală:**

## 3. Aplicarea algoritmilor fundamentali în prelucrarea datelor

Competențe specifice	Conținuturi
<ul style="list-style-type: none"> <li>• Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării</li> <li>• Alegerea unui algoritm eficient de rezolvare a unei probleme</li> </ul>	<p><b>Aplicații interdisciplinare (specifice profilului)</b> Exemple orientative:</p> <ul style="list-style-type: none"> <li>• Rezolvarea ecuației de gradul I și de gradul al II-lea</li> <li>• Amplificarea și simplificarea fracțiilor</li> <li>• Aplicații geometrice (distanța dintre două puncte, aria/perimetrul unui triunghi, volumul corpurilor regulate, centrul de greutate, etc.)</li> <li>• Determinarea primilor n termeni ai unei progresii aritmetice/geometrice</li> <li>• Determinarea punctului de intersecție a două mobile în mișcare rectilinie și uniformă</li> </ul> <p><b>Analiza eficienței unui algoritm.</b></p>

## CONȚINUTURI

### Limbaje de programare

Clasa a IX-a REAL            1 oră pe săptămână

1. Elementele unui limbaj de programare
2. Organizarea fișierelor programelor
3. Sintaxa instrucțiunilor simple (citire, scriere, atribuire), instrucțiunea compusă
4. Parametri și variabile pentru structurile definite de mediul de programare
5. Sintaxa structurilor fundamentale (liniară, alternative, repetitive)
6. Sintaxa și semantica tipurilor de date definite de utilizator
7. Imbricarea structurilor de programare
8. Operații cu fișiere text
9. Eficiența algoritmilor
10. Aplicații de programare avansată

Conținuturile vor fi abordate diferit funcție de nivelul de cunoștințe și specializarea clasei.

Se preconizează ca elevii de la clasele de informatică să administreze (sau să învețe să administreze) sistemele și să le utilizeze la realizarea proiectelor pentru obținerea atestatelor profesionale sau programarea aplicațiilor diverse.

## VALORI ȘI ATITUDINI

- ◆ Dezvoltarea gândirii creative în structurarea și rezolvarea problemelor
- ◆ Formarea obișnuințelor de a recurge la concepte și metode informatice de tip algoritmic specifice în abordarea unei varietăți de probleme.
- ◆ Conștientizarea impactului social, economic și moral al informaticii
- ◆ Manifestarea unor atitudini favorabile față de știință și de cunoaștere în general
- ◆ Manifestarea disponibilității de a evalua/autoevalua activități practice
- ◆ Manifestarea inițiativei și a disponibilității de a aborda sarcini variate

## SUGESTII METODOLOGICE

Pentru buna desfășurare a orelor și aplicarea programei se sugerează următoarele activități de învățare:

- exerciții de scriere identată;
- evidențierea instrumentelor existente pentru realizarea transferului de date între diferite programe;
- exerciții de identificare a activităților care pot fi reprezentate în termeni curenți (din viața de zi cu zi, din disciplinele studiate);
- discuții comparative asupra rezolvărilor problemelor;
- discuții despre avantajele și dezavantajele utilizării unor anumite soft-uri;
- evidențierea greșelilor tipice la declararea variabilelor și la scrierea comenzilor;
- descompunerea unei probleme în entități de prelucrare care se pot rezolva relativ independent;
- reformularea enunțului unei probleme astfel încât să devină transparent modelul identificat;
- asocierea unor tipuri de date și secvențe de operații specifice modelului utilizat;
- desprinderea unor elemente specifice care încadrează problema într-o clasă de probleme similare;
- prezentarea unor probleme care se rezolvă prin descompunere în subprobleme independente și de același tip și recombinația soluțiilor acestora;
- testarea și analizarea comportamentului programelor pentru diferite date de intrare;
- discuții privind validitatea datelor;
- găsirea unor date reprezentative pentru cazuri generale, respectiv descoperirea cazurilor particulare;
- discuții de abordare a problemelor care apar pe parcursul desfășurării activităților;
- evidențierea necesității realizării corecte a unei aplicații;
- prezentarea și dezbaterile aplicațiilor realizate;
- discuții privind validitatea datelor;

- găsirea unor date reprezentative pentru cazuri generale, respectiv descoperirea cazurilor particulare;
- educarea elevilor în ideea că activitatea unui informatician se finalizează cu un produs care trebuie să funcționeze conform condițiilor impuse de beneficiar, să fie însoțit de o documentație, să fie verificat, testat și evaluat ;
- Temele proiectelor vor fi propuse sau alese în colaborare profesor-elev, din domenii diverse, în care se vor folosi cunoștințele dobândite. Proiectul va conține o documentație realizată cu ajutorul unui procesor de text, procesor grafic etc.

## **BIBLIOGRAFIE**

- Tudor Sorin, **INFORMATICĂ**, curs pentru clasele a IX-a și a X-a, profilul real-intensiv, L&S SOFT, 2015.
- Carmen Mincă, Radu Boriga, **Manual de INFORMATICĂ**, clasa a IX-a, profilul real, L&S SOFT, 2014.
- MECI, **Programa școlară de Informatică**, clasa a IX-a, ciclul inferior al liceului, filiera teoretică, profil real, specializarea: Matematică-informatică, intensiv informatică, 2009.