

Anexa nr.la ordinul ministrului educației, cercetării și inovării nr. /.....

MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI INOVĂRII

PROGRAME ȘCOLARE

INFORMATICĂ

CLASA A IX-A

CICLUL INFERIOR AL LICEULUI

*Filiera teoretică, profil real, specializările: Matematică-informatică intensiv informatică
Filiera vocațională, profil militar, specializarea: Matematică-informatică intensiv informatică*

*Aprobată prin ordin al ministrului
Nr. _____ / _____*

București, 2009

NOTA DE PREZENTARE

Prezentul document conține programa școlară pentru disciplina **Informatică**, studiată în filiera teoretică, la profilul real, specializarea matematică-informatică, intensiv informatică, precum și la filiera vocațională, profil militar, specializarea matematică-informatică intensiv informatică, prevăzută săptămânal cu **o oră pentru activități teoretice și 3 ore pentru activități practice**, în conformitate cu **art. 9. din OMECI 3410/16.03.2009**.

Studiul disciplinei Informatică se va desfășura cu întreg colectivul de elevi ai clasei pentru activitățile teoretice și cu **colectivul de elevi organizat pe grupe**, obligatoriu în laboratorul de informatică, pentru activitățile practice.

Programa școlară este parte componentă a curriculumului național. Aceasta reprezintă documentul școlar de tip reglator – instrument de lucru al profesorului – care stabilește, pentru fiecare disciplină, oferta educațională care urmează să fie realizată în bugetul de timp alocat pentru un parcurs școlar determinat, în conformitate cu statutul și locul disciplinei în planul-cadru de învățământ.

Programa școlară pentru învățământul liceal are următoarele componente:

- notă de prezentare
- competențe cheie europene vizate prin studiul disciplinei
- competențe generale
- valori și atitudini
- competențe specifice și conținuturi
- sugestii metodologice.

Nota de prezentare a programei școlare descrie parcursul disciplinei de studiu, argumentează structura didactică adoptată și sintetizează o serie de recomandări considerate semnificative din punct de vedere al finalităților studierii disciplinei respective.

Competențele generale se definesc pentru fiecare disciplină de studiu și au un grad ridicat de generalitate și complexitate.

Valorile și atitudinile orientează dimensiunile axiologică și afectiv-atitudinală aferente formării personalității elevului din perspectiva fiecărei discipline. Realizarea lor concretă derivă din activitatea didactică permanentă a profesorului, constituind un element implicit al acesteia.

Competențele specifice se formează pe parcursul unui an de studiu, sunt deduse din competențele generale și sunt etape în formarea acestora. *Conținuturile învățării* sunt mijloace prin care se urmărește formarea competențelor specifice și implicit a competențelor generale propuse. Unitățile de conținut sunt organizate tematic.

Sugestiile metodologice propun modalități de organizare a procesului de predare-învățare-evaluare. Pentru formarea competențelor specifice pot fi organizate diferite tipuri de activități de învățare. Exemplele de activități de învățare sunt construite astfel încât să pornească de la experiența concretă a elevului și să se integreze unor strategii didactice adecvate contextelor variate de învățare.

În procesul de predare-învățare, activitatea va fi orientată pe probleme: analiza unor situații practice (generale sau specifice unui anumit domeniu), identificarea fluxului informațional, elaborarea unui model algoritmic de rezolvare, implementarea algoritmilor într-un limbaj de programare.

Exemplele utilizate la predare vor fi preponderent alese din aria curriculară a specializării, în colaborare cu profesorii de la aceste discipline.

COMPETENȚELE CHEIE EUROPENE VIZATE PRIN STUDIUL DISCIPLINEI

Pe baza rezultatelor studiilor efectuate la nivelul Comisiei Europene au fost stabilite opt competențe cheie, fiind precizate, pentru fiecare competență cheie, cunoștințele, deprinderile și atitudinile care trebuie dobândite, respectiv formate elevilor în procesul educațional.

Aceste competențe cheie răspund obiectivelor asumate pentru dezvoltarea sistemelor educaționale și de formare profesională în Uniunea Europeană și, ca urmare, stau la baza stabilirii curriculumului pentru educația de bază.

Principalele competențe cheie europene vizate prin studiul disciplinei sunt:

Competențe în matematică și competențe de bază în științe și tehnologie
Competențe digitale

COMPETENȚE GENERALE

1. Identificarea conexiunilor dintre informatică și societate
2. Identificarea datelor care intervin într-o problemă și a relațiilor dintre acestea
3. Elaborarea algoritmilor de rezolvare a problemelor
4. Aplicarea algoritmilor fundamentali în prelucrarea datelor
5. Implementarea algoritmilor într-un limbaj de programare

VALORI ȘI ATITUDINI

1. Exprimarea unui mod de gândire creativ, în structurarea și rezolvarea problemelor
2. Conștientizarea impactului social, economic și moral al informaticii
3. Formarea obișnuințelor de a recurge la concepte și metode informatice de tip algoritmic specifice în abordarea unei varietăți de probleme
4. Manifestarea unor atitudini favorabile față de știință și de cunoaștere în general
5. Manifestarea inițiativei și disponibilității de a aborda sarcini variate

COMPETENȚE SPECIFICE ȘI CONȚINUTURI

1. Identificarea conexiunilor dintre informatică și societate

Competențe specifice	Conținuturi ¹
1.1. Identificarea aplicațiilor informaticii în viața socială 1.2. Recunoașterea situațiilor în care este necesară prelucrarea algoritmică a informațiilor.	Definirea informaticii ca știință Rolul informaticii în societate Studii de caz ale unor situații sociale, în abordare informatizată

2. Identificarea datelor care intervin într-o problemă și a relațiilor dintre acestea

Competențe specifice	Conținuturi
2.1. Descrierea unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire 2.2. Descrierea unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire	Date cu care lucrează algoritmi (constante, variabile, expresii). Operații asupra datelor (aritmetice, logice, relaționale).

3. Elaborarea algoritmilor de rezolvare a problemelor

Competențe specifice	Conținuturi
3.1. Analizarea enunțului unei probleme și stabilirea pașilor de rezolvare a problemei. 3.2. Reprezentarea algoritmilor în pseudocod. 3.3. Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.	<p>Etapele rezolvării problemelor. Exemple.</p> <p>Noțiunea de algoritm. Caracteristici. Exemple.</p> <p>Reprezentarea algoritmilor. Pseudocod. Principiile programării structurate. Structuri de bază: structura liniară, structura alternativă, structura repetitivă.</p> <p>Algoritmi elementari.</p> <p>1. Prelucrarea numerelor :</p> <ul style="list-style-type: none"> • prelucrarea cifrelor unui număr (de exemplu, suma cifrelor, testarea proprietății de palindrom, etc.) • probleme de divizibilitate (de exemplu, determinarea divizorilor unui număr, determinarea c.m.m.d.c./c.m.m.m.c., testare primalitate, etc.) • calculul unor expresii simple (sume, produse, etc.) <p>2. Prelucrarea unor secvențe de valori:</p> <ul style="list-style-type: none"> • determinare minim/maxim • verificarea unei proprietăți (de exemplu, dacă toate elementele din secvență sunt numere perfecte, etc.) • calculul unor expresii în care intervin valori din secvență (de exemplu: numărarea elementelor pare/impare, etc.) • generarea șirurilor recurente (de exemplu: șirul Fibonacci)

¹ Conținuturile sunt prezentate în tabele, grupate pe competențe și asocierea acestora este obligatorie. Este la decizia cadrului didactic/ a autorului de manual școlar ordinea abordării conținuturilor, cu respectarea logicii interne a domeniului.

4. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice	Conținuturi
<p>4.1. Transcrierea algoritmilor din pseudocod într-un limbaj de programare.</p> <p>4.2. Identificarea necesității structurării datelor în tablouri.</p> <p>4.3. Prelucrarea datelor structurate .</p> <p>4.4. Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor.</p> <p>4.5. Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++)</p>	<p>Elementele de bază ale limbajului de programare</p> <p>Noțiuni introductive</p> <ul style="list-style-type: none"> • Structura programelor • Vocabularul limbajului • Tipuri simple de date (standard) • Constante, variabile, expresii • Citirea/scrierea datelor <p>Structuri de control</p> <ul style="list-style-type: none"> • Structura liniară • Structura alternativă • Structuri repetitive <p>Tipuri structurate de date. Tipul tablou</p> <ul style="list-style-type: none"> • Tablouri unidimensionale • Tablouri bidimensionale <p>Algoritmi fundamentali de prelucrare a datelor structurate în tablouri</p> <ul style="list-style-type: none"> • căutare secvențială, căutare binară • sortare • interclasare • parcurgerea tablourilor bidimensionale pe linii/coloane <p>Fișiere text. Definire, operații specifice</p> <p>Mediul limbajului de programare studiat</p> <ul style="list-style-type: none"> • Prezentare generală • Editarea programelor sursă • Compilare, rulare, depanare

5. Aplicarea algoritmilor fundamentali în prelucrarea datelor

Competențe specifice	Conținuturi
<p>5.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării</p> <p>5.2. Alegerea unui algoritm eficient de rezolvare a unei probleme</p>	<p>Aplicații interdisciplinare (specifice profilului)</p> <p>Exemple orientative:</p> <ul style="list-style-type: none"> • Rezolvarea ecuației de gradul I și de gradul al II-lea • Simplificarea fracțiilor • Aplicații geometrice (distanța dintre două puncte, aria/perimetrul unui triunghi, volumul corpurilor regulate, centrul de greutate al unei mulțimi de puncte, etc.) • Determinarea primilor n termeni ai unei progresii aritmetice/geometrice • Determinarea punctului de intersecție a două mobile în mișcare rectilinie și uniformă • Determinarea masei moleculare a unui compus chimic <p>Analiza eficienței unui algoritm.</p>

SUGESTII METODOLOGICE

Predarea informaticii va fi orientată pe *rezolvarea de probleme*, utilizându-se preponderent metode activ-participative și punându-se accent pe *analiza problemei*. Pentru buna desfășurare a orelor și aplicarea programei se sugerează următoarele activități de învățare:

- discuții despre activități cotidiene și modelarea acestora sub forma unei secvențe bine definite de pași;
- combinarea unor operații elementare (pași) pentru obținerea anumitor prelucrări complexe în funcție de scopul propus;
- explicarea conceptului de algoritm și a caracteristicilor algoritmilor;
- explicarea diferenței existente între informații care reprezintă date și cele care descriu calea de rezolvare a unei probleme;
- descrierea unui algoritm în limbaj natural;
- prezentarea obiectelor cu care operează algoritmi (constante, variabile, expresii);
- clasificarea datelor în date de intrare, de ieșire și date de manevră;
- descrierea etapelor rezolvării unei probleme din punct de vedere algoritmic;
- prezentarea schemelor logice ca modalitate de reprezentare a algoritmilor;
- prezentarea structurilor de bază în pseudocod;
- exersarea scrierii unor algoritmi simpli, folosind structuri liniare, alternative și repetitive;
- identificarea unor situații în care alegerea unui algoritm prezintă avantaje în raport cu altul;
- exersarea creării și aplicării algoritmilor pentru rezolvarea unor probleme întâlnite de elevi în studiul altor discipline școlare;
- evidențierea greșelilor tipice în elaborarea algoritmilor;
- prezentarea unor exemple de implementare într-un limbaj de programare a unor algoritmi elaborați de elevi și executarea acestora pe calculator;
- prezentarea mediului de programare (facilități de editare, de compilare și de rulare);
- prezentarea și exemplificarea elementelor de bază ale limbajului de programare;
- activități de dezvoltare a deprinderilor de organizare a informațiilor în tablouri;
- exersarea citirii datelor de la tastatură și a afișării datelor pe ecran;
- exersarea scrierii unor programe simple;
- codificarea structurilor de control învățate în limbaj de programare;
- proiectarea unui algoritm și implementarea acestuia;
- folosirea facilităților mediului în depanarea programelor;
- implementarea structurilor de tip tablou;
- exerciții de transfer al datelor din/în fișiere text;
- evidențierea analogiilor și diferențelor între citirea/scrierea utilizând dispozitivele standard de intrare/ieșire și fișiere text;
- testarea și analizarea comportamentului programelor pentru diferite date de intrare;
- încurajarea discuțiilor purtate între elevi, exprimarea și ascultarea părerilor fiecăruia.