

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

## **MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI INOVĂRII**

### **PROGRAME ȘCOLARE**

### **INFORMATICĂ**

### **CLASA A XI-A**

### **CICLUL SUPERIOR AL LICEULUI**

*Filiera teoretică, profil real, specializarea: Matematică-informatică*

*Filiera vocațională, profil militar, specializarea: Matematică-informatică*

**Aprobată prin ordin al ministrului**

**Nr. \_\_\_\_\_ / \_\_\_\_\_**

**București, 2009**

## NOTA DE PREZENTARE

Prezentul document cuprinde programa școlară de *Informatică*, clasa a XI-a, pentru specializarea matematică-informatică din cadrul filierelor teoretică, profilul real, și vocațională, profil militar MApN.

În conformitate cu planurile-cadru de învățământ pentru ciclul superior al liceului, aprobate prin ordinul ministrului educației și cercetării nr. 3410/ 16.03.2009, la această specializare se alocă **4 ore pe săptămână** pentru studierea disciplinei *Informatică*. Cele 4 ore vor fi distribuite astfel: **o oră de teorie și 3 ore activități practice desfășurate în laborator.**

**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.

**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 elaborarea prezentei programei școlare au fost respectate principiile de proiectare curriculară, specifice curriculumului național, valorificându-se în același timp tendințele domeniului pe plan internațional și opinii ale unor profesori cu o bogată experiență didactică.

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 de la disciplinele din aria curriculară *Matematică și științe ale naturii*, 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 competențele cheie europene vizate prin studiul disciplinei sunt:

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

## COMPETENȚE GENERALE

1. Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora
2. Elaborarea algoritmilor de rezolvare a problemelor
3. 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 datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora

| Competențe specifice  | Conținuturi <sup>1</sup>  |
|---|---|
| <p>1.1. Analizarea unei probleme în scopul identificării datelor necesare și alegerea modalităților adecvate de structurare a datelor care intervin într-o problemă</p> <p>1.2. Identificarea avantajelor utilizării diferitelor metode de structurare a datelor necesare pentru rezolvarea unei probleme</p> <p>1.3. Utilizarea șirurilor de caractere și a structurilor de date neomogene în modelarea unor situații problemă</p> <p>1.4. Implementarea unor algoritmi de prelucrare a a șirurilor de caractere și a structurilor neomogene</p> <p>1.5. Transpunerea unei probleme din limbaj natural în limbaj de grafuri, folosind corect terminologia specifică</p> <p>1.6. Descrierea unor algoritmi simpli de verificare a unor proprietăți specifice grafurilor</p> | <p><b>Tablouri bidimensionale*</b></p> <p><b>Șiruri de caractere</b></p> <ul style="list-style-type: none"> <li>• Particularități de memorare a șirurilor de caractere</li> <li>• Subprograme predefinite de prelucrare a șirurilor de caractere</li> </ul> <p><b>Structuri de date neomogene (struct/record)</b></p> <ul style="list-style-type: none"> <li>• Rezolvarea unor probleme cu caracter practic.</li> </ul> <p><b>Liste</b></p> <ul style="list-style-type: none"> <li>• Reprezentarea grafică a structurilor de tip listă</li> <li>• Operații specifice</li> <li>• Stiva și coada</li> <li>• Aplicații cu implementare statică</li> </ul> <p><b>Grafuri orientate și neorientate</b></p> <ul style="list-style-type: none"> <li>• Terminologie și proprietăți (graf orientat și neorientat, adiacență, incidență, grad; lanț, lanț elementar, drum, drum elementar, ciclu, ciclu elementar, circuit, circuit elementar; subgraf, graf parțial; conexitate, tare conexitate, arbore, arbore parțial, etc)</li> <li>• Reprezentarea în memorie a grafurilor (matrice de adiacență, liste de adiacență, lista muchiilor/arcelor)</li> <li>• Parcurgerea grafurilor. Aplicații.</li> </ul> |

### 2. Elaborarea algoritmilor de rezolvare a problemelor

| Competențe specifice   | Conținuturi  |
|--|--|
| <p>2.1 Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei</p> <p>2.2 Construirea unor soluții pentru probleme simple care se rezolvă cu ajutorul metodelor de programare</p> <p>2.3 Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme intradisciplinare sau interdisciplinare, sau a unor probleme cu aplicabilitate practică</p> | <p><b>Metoda de programare <i>Divide et Impera</i></b></p> <ul style="list-style-type: none"> <li>• Prezentare generală</li> <li>• Aplicații</li> </ul> <p><b>Metoda de programare <i>Backtracking</i></b></p> <ul style="list-style-type: none"> <li>• Prezentare generală</li> <li>• Implementarea iterativă sau recursivă a algoritmilor de generare a produsului cartezian, permutărilor, combinațiilor, aranjamentelor, submulțimilor unei mulțimi</li> </ul> |

<sup>1</sup> 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.

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

| Competențe specifice   | Conținuturi   |
|--|---|
| 3.1 Recunoașterea situațiilor în care este necesară utilizarea unor subprograme                                | <b>Subprograme</b> <ul style="list-style-type: none"><li>• Structura și a modul de definire al subprogramelor</li><li>• Declararea și apelul subprogramelor</li><li>• Transferul parametrilor la apel (prin valoare și referință)</li><li>• Returnarea valorilor de către subprograme</li><li>• Variabile locale și globale</li><li>• Aplicații folosind subprograme</li></ul> <b>Recursivitate</b> <ul style="list-style-type: none"><li>• Definiție. Exemplificare</li><li>• Mecanisme de implementare</li><li>• Aplicații cu subprograme recursive</li></ul> |
| 3.2 Analiza problemei în scopul identificării subproblemelor acesteia  |   |
| 3.3 Utilizarea corectă a subprogramelor predefinite și a celor definite de utilizator                          |   |
| 3.4 Descrierea și aplicarea mecanismului recursivității  |   |
| 3.5 Identificarea avantajelor și a dezavantajelor aplicării tehnicii recursive în implementarea unor rezolvări |   |

\* Se studiază în vederea corelării programei curente cu programa anterioară.

#### 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 în limbaj informatic;
- activități de dezvoltare a deprinderilor de organizare a informației în diferite structuri de date (șiruri de caractere, articole, liste alocate static, stive și cozi);
- identificarea modalităților eficiente de reprezentare a datelor necesare pentru rezolvarea unei probleme
- descompunerea rezolvării unei probleme în subprobleme;
- prezentarea unor situații practice familiare elevilor care pot fi modelate în termenii teoriei grafurilor
- reprezentarea grafică a grafurilor, listelor și ilustrarea prin exemple reprezentate grafic a diferitelor noțiuni și proprietăți specifice
- demonstrarea modului de realizare a operațiilor elementare specifice diferitelor structuri de date pe exemple reprezentate grafic.
- aplicarea, pe exemple relevante, a algoritmilor fundamentali din teoria grafurilor
- adaptarea creativă a algoritmilor fundamentali de prelucrare a datelor pentru rezolvarea unei probleme
- identificarea unor situații în care alegerea unui algoritm prezintă avantaje în raport cu altul;
- exersarea creării și aplicării programelor pentru rezolvarea unor probleme întâlnite de elevi în studiul altor discipline școlare;
- evidențierea greșelilor tipice în elaborarea algoritmilor;
- proiectarea/modelarea unor algoritmi și implementarea acestora;
- testarea și analiza comportamentului programelor pentru diferite date de intrare;
- încurajarea discuțiilor purtate între elevi, exprimarea și ascultarea părerilor fiecăruia.

Conținuturile din prezenta programă vor fi susținute prin rezolvarea unor probleme intradisciplinare sau interdisciplinare, respectiv probleme cu aplicabilitate practică în viața cotidiană.

Elevii care manifestă interes pentru grafică pe calculator pot fi încurajați să realizeze prezentări în PowerPoint, Flash sau alte programe, care să illustreze teoria și aplicații ale acesteia.

## Exemple de aplicații recomandate pentru unele conținuturi:

### Grafuri orientate și grafuri neorientate

1. Algoritmi simpli de verificare a însușirii terminologiei sau de verificare a unor proprietăți specifice grafurilor (de exemplu, calcularea gradelor vârfurilor unui graf, verificarea faptului că o succesiune de vârfuri reprezintă un lanț, drum, ciclu sau circuit în graf etc.)

Probleme practice, care solicită aplicarea algoritmilor de parcurgere a grafurilor (conexitatea unei structuri relaționale între mai multe persoane, puncte accesibile ale unei rețele de comunicație etc.)

### Recursivitate

1. Exemplificarea recursivității prin recurențe matematice și modele intuitive
2. Compararea implementării recursive cu cea iterativă în alegerea algoritmului de rezolvare a unei probleme

### Metoda de programare *Divide et Impera*

1. Sortarea eficientă a unei mulțimi de valori aplicând metoda *Divide et Impera* (sortarea rapidă sau sortarea prin interclasare)
2. Căutarea eficientă a unui element într-o mulțime ordonată aplicând metoda *Divide et Impera* (căutarea binară)

### Metoda de programare *Backtracking*

1. Colorarea unei hărți
2. Problema celor n dame
3. Plata unei sume folosind monede de valori date