

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
CONSILIUL NAȚIONAL PENTRU CURRICULUM

PROGRAME ȘCOLARE PENTRU CLASA A X-A
CICLUL INFERIOR AL LICEULUI

INFORMATICĂ
(CURRICULUM DIFERENȚIAT)

Profil real

Specializările: Matematică-informatică
Științe ale naturii

Aprobat prin ordin al ministrului
Nr. 4598 / 31.08.2004

București, 2004

NOTA DE PREZENTARE

Prezentul curriculum școlar cuprinde programa pentru disciplina *Informatică*, studiată în filiera teoretică, la profilul real, specializările matematică-informatică și științele naturii, prevăzută cu o oră pe săptămână în componenta curriculum diferențiat.

Programa are următoarele componente:

- Notă de prezentare
- Competențe generale
- Competențe specifice și conținuturi
- Valori și atitudini
- Sugestii metodologice.

În elaborarea programei de față au fost luate în considerație atât cercetările în domeniul curricular, tendințe pe plan internațional, cât și opiniile unor profesori cu o bogată experiență didactică.

Ținând cont de noutatea formei de prezentare, sunt necesare următoarele completări:

- **Competențele generale** sunt ansambluri structurate de cunoștințe și deprinderi; ele se definesc pe obiect de studiu și se formează pe durata unui ciclu curricular.
- **Competențele specifice** se definesc pe obiect de studiu și se formează pe durata unui an de studiu. Ele sunt derivate din competențele generale.

Studiul disciplinei *Informatică* se va desfășura cu întreg colectivul de elevi ai clasei.

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.

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

COMPETENȚE GENERALE

Identificarea conexiunilor dintre informatică și societate.

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

Elaborarea algoritmilor de rezolvare a problemelor

Aplicarea algoritmilor fundamentali în prelucrarea datelor

Implementarea algoritmilor într-un limbaj de programare

VALORI ȘI ATITUDINI

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

COMPETENȚE SPECIFICE ȘI CONȚINUTURI

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

Competențe specifice	Conținuturi
<ol style="list-style-type: none"> 1. Implementarea algoritmilor reprezentați în pseudocod în limbaj de programare 2. Utilizarea unui mediu de programare (pentru limbajul Pascal sau pentru limbajul C/C++) 	<p>Elementele de bază ale limbajului de programare</p> <ol style="list-style-type: none"> 1. Noțiuni introductive <ul style="list-style-type: none"> - Structura programelor - Vocabularul limbajului - Tipuri simple de date (standard) - Constante, variabile, expresii - Citirea/scrierea datelor 2. Structuri de control <ul style="list-style-type: none"> - Structura liniară - Structura alternativă - Structuri repetitive 3. Mediul limbajului de programare studiat <ul style="list-style-type: none"> - Prezentare generală - Editarea programelor sursă - Compilare, rulare, depanare 4. Implementarea unor algoritmi elementari elaborați în clasa a IX-a (aplicații interdisciplinare).

Implementarea algoritmilor într-un limbaj de programare

Competențe specifice	Conținuturi
<ol style="list-style-type: none"> 2.1. Identificarea necesității structurării datelor în tablouri 2.2. Prelucrarea datelor structurate în tablouri unidimensionale. 2.3. Utilizarea fișierelor text pentru introducerea datelor și extragerea rezultatelor 	<ol style="list-style-type: none"> 5. Tipuri structurate de date. Tipul tablou. Tablouri unidimensionale 6. Fișiere text. Definire, operații specifice

Elaborarea algoritmilor de rezolvare a problemelor

Competențe specifice	Conținuturi
<ol style="list-style-type: none"> 3.1. Prelucrarea datelor structurate în tablouri unidimensionale. 	<ol style="list-style-type: none"> 7. Algoritmi fundamentali de prelucrare a datelor structurate în tablouri <ul style="list-style-type: none"> - căutare secvențială, căutare binară - sortare - interclasare

Aplicarea algoritmilor fundamentali în prelucrarea datelor

Competențe specifice	Conținuturi
<ol style="list-style-type: none"> 4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 4.2. Alegerea celui mai eficient algoritm de rezolvare a unei probleme 	<p>Aplicații interdisciplinare (specifice profilului).</p> <p>Exemple orientative:</p> <ul style="list-style-type: none"> - Prelucrări statistice ale unei serii de valori - Operații cu polinoame - Calcule combinatoriale - Determinarea unor mărimi fizice dintr-un circuit electric serie/paralel - Aplicații din genetică (legea creșterilor organice, etc.) <p>Analizarea eficienței unui algoritm.</p>

Competențe specifice	Conținuturi
5.1. Identificarea aplicațiilor informaticii în viața socială	Aplicații din viața cotidiană Exemple orientative:
5.2. Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene	- Determinarea situației școlare a unui elev (medii semestriale, medii generale, numărul de absențe, etc.) - Balanța de cheltuieli ale unei familii - Determinarea salariului unei persoane - Evidența operațiilor într-un cont bancar

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:

- ♦ modelarea unor activități cotidiene cu ajutorul instrumentelor informatice
- ♦ combinarea unor operații elementare (pași) pentru obținerea anumitor prelucrări complexe în funcție de scopul propus;
- ♦ descrierea unui algoritm în limbaj natural, ca etapă inițială în elaborarea programului
- ♦ descrierea detaliată a etapelor rezolvării unei probleme din punct de vedere algoritmic;
- ♦ compararea unor algoritmi de rezolvare a unei probleme, în scopul alegerii algoritmului eficient;
- ♦ exersarea creării și aplicării algoritmilor pentru rezolvarea unor probleme întâlnite de elevi în studiul altor discipline școlare sau în viața cotidiană;
- ♦ prezentarea mediului de programare (facilități de editare, de compilare și de rulare);
- ♦ familiarizarea elevului cu noțiunea de limbaj de programare;
- ♦ prezentarea și exemplificarea elementelor de bază ale limbajului de programare;
- ♦ activități de formare a deprinderilor de organizare a informațiilor în tablouri unidimensionale, prin exemplificări concludente;
- ♦ utilizarea intrării și ieșirii standard;
- ♦ exersarea scrierii unor programe simple;
- ♦ implementarea structurilor de control învățate;
- ♦ exerciții de transpunere a pașilor unui algoritm în structuri de control specifice;
- ♦ proiectarea/modelarea unui algoritm și implementarea acestuia;
- ♦ folosirea facilităților mediului în depanarea programelor;
- ♦ prezentarea structurilor de date standard;
- ♦ implementarea structurilor de tip tablou;
- ♦ exerciții de transfer al datelor din/în fișiere text;
- ♦ evidențierea analogiilor și diferențelor dintre citirea/scrierea utilizând dispozitivele standard de intrare/ieșire și fișiere text;
- ♦ testarea și analizarea comportamentului programelor pentru diferite date de intrare;
- ♦ evidențierea greșelilor tipice în elaborarea programelor;
- ♦ încurajarea discuțiilor purtate între elevi, exprimarea și ascultarea părerilor fiecăruia.

Datorită caracterului pregnant aplicativ al disciplinei informatică, se recomandă desfășurarea orelor în laboratorul de informatică.

??????

Identificarea conexiunilor dintre informatică și societate.

Competențe specifice	Conținuturi
5.1. Identificarea aplicațiilor informaticii în viața socială 5.2. Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene	<ul style="list-style-type: none">• Aplicații din viața cotidiană• Exemple orientative:<ul style="list-style-type: none">• Determinarea situației școlare a unei clase de elevi (medii semestriale, medii generale, numărul de absențe, etc.)• Evidența cheltuielilor la întreținere pentru locatarii unui bloc• Evidența operațiilor cu conturi bancare.

Aplicarea algoritmilor fundamentali în prelucrarea datelor

Competențe specifice	Conținuturi
4.3. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 4.4. Alegerea celui mai eficient algoritm de rezolvare a unei probleme	<p>Aplicații interdisciplinare (specifice profilului).</p> <p>Exemple orientative:</p> <ul style="list-style-type: none">- Prelucrări statistice ale unei serii de valori- Operații cu polinoame- Calcule combinatoriale- Determinarea unor mărimi fizice dintr-un circuit electric serie/paralel- Aplicații din genetică (legea creșterilor organice, etc.) <p>Analizarea eficienței unui algoritm.</p>

Identificarea conexiunilor dintre informatică și societate.

Competențe specifice	Conținuturi
5.3. Identificarea aplicațiilor informaticii în viața socială 5.4. Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene	<p>Aplicații din viața cotidiană</p> <p>Exemple orientative:</p> <ul style="list-style-type: none">- Determinarea situației școlare a unui elev (medii semestriale, medii generale, numărul de absențe, etc.)- Balanța de cheltuieli ale unei familii- Determinarea salariului unei persoane- Evidența operațiilor într-un cont bancar

<http://www.edu.ro/pr102005.htm>

***Anexa nr. 2 la ordinul ministrului educației și cercetării nr. 4598 /
31.08.2004***

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII

CONSILIUL NAȚIONAL PENTRU CURRICULUM

PROGRAME ȘCOLARE PENTRU CLASA A X-A
CICLUL INFERIOR AL LICEULUI

INFORMATICĂ

Profil real

Specializarea: Matematică-informatică, intensiv informatică

*Aprobat prin ordin al ministrului
Nr. 4598 / 31.08.2004*

București, 2004

NOTA DE PREZENTARE

Prezentul curriculum școlar cuprinde programa pentru disciplina *Informatică*, studiată în filiera teoretică, la profilul real, specializarea matematică-informatică, intensiv informatică, prevăzută săptămânal cu două ore pentru activități teoretice și două ore pentru activități practice, în conformitate cu art. 9. alin c) din OMECT nr. 5723/ 23.12.2003.

Programa are următoarele componente:

- ♦ Notă de prezentare
- ♦ Competențe generale
- ♦ Competențe specifice și conținuturi
- ♦ Valori și atitudini
- ♦ Sugestii metodologice.

În elaborarea programei de față au fost luate în considerație atât cercetările în domeniul curricular, tendințe pe plan internațional, cât și opiniile unor profesori cu o bogată experiență didactică.

Ținând cont de noutatea formei de prezentare, sunt necesare următoarele completări:

- ♦ **Competențele generale** sunt ansambluri structurate de cunoștințe și deprinderi; ele se definesc pe obiect de studiu și se formează pe durata unui ciclu curricular.
- ♦ **Competențele specifice** se definesc pe obiect de studiu și se formează pe durata unui an de studiu. Ele sunt derivate din competențele generale.

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

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.

COMPETENȚE GENERALE

Identificarea conexiunilor dintre informatică și societate.

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

Elaborarea algoritmilor de rezolvare a problemelor

Aplicarea algoritmilor fundamentali în prelucrarea datelor

Implementarea algoritmilor într-un limbaj de programare

VALORI ȘI ATITUDINI

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

COMPETENȚE SPECIFICE ȘI CONȚINUTURI

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

Competențe specifice	Conținuturi
1.1. Evidențierea necesității structurării datelor 1.2. Prelucrarea datelor structurate 1.3. Alegerea structurii de date adecvate rezolvării unei probleme	1. Tipuri structurate de date. ♦ Înregistrare (structură) ♦ Șir de caractere ♦ Listă, stivă, coadă

4. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice	Conținuturi
2.1. Utilizarea corectă a subprogramelelor predefinite și a celor definite de utilizator 2.2. Construirea unor subprograme pentru rezolvarea subproblemelor unei probleme 2.3. Aplicarea mecanismului recursivității prin crearea unor subprograme recursive (definite de utilizator) 2.4. Compararea dintre implementarea recursivă și cea iterativă a aceluiași algoritm	1. Subprograme 2. Declararea, definirea și apelul subprogramelelor 3. Transferul parametrilor la apel 4. Returnarea valorilor de către subprograme 5. Variabile locale și globale 6. Subprograme recursive. Mecanismul de realizare a recursivității

5. Elaborarea algoritmilor de rezolvare a problemelor

Competențe specifice	Conținuturi
3.1. Prelucrarea datelor structurate 3.2. Recunoașterea situațiilor în care este necesară utilizarea unor subprograme 3.3. Analiza problemei în scopul identificării subproblemelor acesteia 3.4. Descrierea metodei de rezolvare a unei probleme în termeni recursivi	<ul style="list-style-type: none"> Algoritmi fundamentali de prelucrare a datelor structurate prelucrarea unui șir de caractere la nivel de caracter și la nivel de structură, utilizând funcții specifice prelucrarea unei înregistrări/structuri la nivel de câmp și la nivel de structură Aplicații folosind subprograme

6. Aplicarea algoritmilor fundamentali în prelucrarea datelor

Competențe specifice	Conținuturi
4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 4.2. Alegerea celui mai eficient algoritm de rezolvare a unei probleme	<ul style="list-style-type: none"> Aplicații interdisciplinare (specifice profilului) cu structuri de date și subprograme. Exemple orientative: <ul style="list-style-type: none"> Prelucrări statistice ale unei serii de valori Operații cu polinoame Calcul și generări combinatoriale Determinarea unor mărimi fizice dintr-un circuit electric serie/paralel Aplicații din genetică (legea creșterilor organice, etc.) Prelucrarea de text (căutarea unui subșir într-un șir, transformări, delimitări cuvinte, etc.) Analizarea eficienței unui algoritm.

7. Identificarea conexiunilor dintre informatică și societate.

Competențe specifice	Conținuturi
5.3. Identificarea aplicațiilor informaticii în viața socială 5.4. Elaborarea și implementarea unor algoritmi de rezolvare a unor probleme cotidiene	<ul style="list-style-type: none">• Aplicații din viața cotidiană• Exemple orientative:• Determinarea situației școlare a unei clase de elevi (medii semestriale, medii generale, numărul de absențe, etc.)• Evidența cheltuielilor la întreținere pentru locatarii unui bloc• Evidența operațiilor cu conturi bancare.

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 prelucrări elementare pentru obținerea anumitor prelucrări complexe în funcție de scopul propus;
- ♦ explicarea conceptelor referitoare la subprograme;
- ♦ descompunerea rezolvării unei probleme în subprobleme;
- ♦ 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;
- ♦ activități de dezvoltare a deprinderilor de organizare a informației în diferite structuri de date;
- ♦ exersarea definirii și apelării unor programe simple;
- ♦ proiectarea/modelarea unor algoritmi și implementarea acestora;
- ♦ implementarea structurilor de tip sir de caractere, înregistrare (structura);
- ♦ testarea și analizarea comportamentului programelor pentru diferite date de intrare;
- ♦ încurajarea discuțiilor purtate între elevi, exprimarea și ascultarea părerilor fiecăruia.

PROGRAME ȘCOLARE PENTRU CLASA A XI-A

CICLUL SUPERIOR AL LICEULUI

INFORMATICĂ

Profil real

Specializarea: Matematică-informatică

București, 2005

NOTA DE PREZENTARE

Prezentul curriculum școlar cuprinde programa pentru disciplina *Informatică*, studiată în filiera teoretică, la profilul real, specializarea matematică-informatică, intensiv informatică, prevăzută săptămânal cu **o oră** pentru activități teoretice **și trei ore** pentru activități practice, în conformitate **cu art. 9. alin c) din OMECT nr. 5723/ 23.12.2003.**

Programa are următoarele componente:

- Notă de prezentare
- Competențe generale
- Competențe specifice și conținuturi
- Valori și atitudini
- Sugestii metodologice.

În elaborarea programei de față au fost luate în considerație atât cercetările în domeniul curricular, tendințe pe plan internațional, cât și opiniile unor profesori cu o bogată experiență didactică.

Ținând cont de noutatea formei de prezentare, sunt necesare următoarele completări:

- **Competențele generale** sunt ansambluri structurate de cunoștințe și deprinderi; ele se definesc pe obiect de studiu și se formează pe durata unui ciclu curricular.
- **Competențele specifice** se definesc pe obiect de studiu și se formează pe durata unui an de studiu. Ele sunt derivate din competențele generale.

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.

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 științele naturii, în colaborare cu profesorii de la aceste discipline.

COMPETENȚE GENERALE

Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora

Elaborarea algoritmilor de rezolvare a problemelor

Implementarea algoritmilor într-un limbaj de programare

VALORI ȘI ATITUDINI

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

COMPETENȚE SPECIFICE ȘI CONȚINUTURI

8. Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora

Competențe specifice	Conținuturi
<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 tablourilor bidimensionale, a șirurilor de caractere și a structurilor de date neomogene în modelarea unor situații problemă</p> <p>1.4. Implementarea unor algoritmi de prelucrare a tablourilor bidimensionale, 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>1. Tablouri bidimensionale</p> <ul style="list-style-type: none"> • Terminologie (indici, linie, coloană, matrice pătrată, diagonală principală, diagonală secundară) • Algoritmi de prelucrare a tablourilor bidimensionale <p>2. Șiruri de caractere</p> <ul style="list-style-type: none"> • Particularități de memorare a șirurilor de caractere • Subprograme predefinite de prelucrare a șirurilor de caractere <p>3. Structuri de date neomogene (struct/record)</p> <ul style="list-style-type: none"> • Rezolvarea unor probleme cu caracter practic. <p>4. Liste</p> <ul style="list-style-type: none"> • Reprezentarea grafică a structurilor de tip listă • Operații specifice • Stiva și coada • Aplicații cu implementare statică <p>5. Grafuri orientate și neorientate</p> <ul style="list-style-type: none"> • Terminologie și proprietăți (graf orientat și neorientat, adiacență, incidență, grad; lanț, drum, ciclu, circuit; subgraf, graf parțial; conexitate, arbore, arbore parțial) • Reprezentarea în memorie a grafurilor (matrice de adiacență, liste de adiacență, lista muchiilor/arcilor) • Parcurgerea grafurilor. Aplicații.

9. 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 metodei <i>backtracking</i></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>1. Metoda de programare <i>Divide et Impera</i></p> <ul style="list-style-type: none"> • Prezentare generală • Aplicații <p>2. Metoda de programare <i>Backtracking</i></p> <ul style="list-style-type: none"> • Prezentare generală • Implementarea iterativă sau recursivă a algoritmilor de generare a produsului cartezian, permutărilor, combinațiilor, aranjamentelor, submulțimilor unei mulțimi

10. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice	Conținuturi
3.1 Recunoașterea situațiilor în care este necesară utilizarea unor subprograme 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 dezavantajelor aplicării tehnicii recursive în implementarea unor rezolvări	1. Subprograme <ul style="list-style-type: none">• Structura și a modul de definire al subprogramelor• Declararea și apelul subprogramelor• Transferul parametrilor la apel (prin valoare și prin adresă)• Returnarea valorilor de către subprograme• Variabile locale și globale• Aplicații folosind subprograme 2. Recursivitate. <ul style="list-style-type: none">• Definiție. Exemplificare• Mecanisme de implementare• Aplicații cu subprograme recursive

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 algoritmic;
- activități de dezvoltare a deprinderilor de organizare a informației în diferite structuri de date (tablouri, 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 algoritmilor fundamentali din teoria grafurilor pe exemple relevante
- 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 analizarea comportamentului programelor pentru diferite date de intrare;
- încurajarea discuțiilor purtate între elevi, exprimarea și ascultarea părerilor fiecăruia.

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

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 lanț, drum, ciclu sau circuit în graf etc.)
2. 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.)

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)

Recursivitate

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

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII
CONSILIUL NAȚIONAL PENTRU CURRICULUM

PROGRAME ȘCOLARE PENTRU CLASA A XI-A
CICLUL SUPERIOR AL LICEULUI

INFORMATICĂ

Profil real

Specializarea: Matematică-informatică, intensiv informatică

București, 2005

Prezentul curriculum școlar cuprinde programa pentru disciplina *Informatică*, studiată în filiera teoretică, la profilul real, specializarea matematică-informatică, intensiv informatică, prevăzută **săptămânal cu trei ore pentru activități teoretice și patru ore pentru activități practice, în conformitate cu art. 9. alin c) din OMECT nr. 5723/ 23.12.2003.**

Programa are următoarele componente:

- ♦ Notă de prezentare
- ♦ Competențe generale
- ♦ Competențe specifice și conținuturi
- ♦ Valori și atitudini
- ♦ Sugestii metodologice.

În elaborarea programei de față au fost luate în considerație atât cercetările în domeniul curricular, tendințe pe plan internațional, cât și opiniile unor profesori cu o bogată experiență didactică.

Ținând cont de noutatea formei de prezentare, sunt necesare următoarele completări:

- ♦ **Competențele generale** sunt ansambluri structurate de cunoștințe și deprinderi; ele se definesc pe obiect de studiu și se formează pe durata unui ciclu curricular.
- ♦ **Competențele specifice** se definesc pe obiect de studiu și se formează pe durata unui an de studiu. Ele sunt derivate din competențele generale.

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.

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.

COMPETENȚE GENERALE

Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora

Elaborarea algoritmilor de rezolvare a problemelor

Implementarea algoritmilor într-un limbaj de programare

VALORI ȘI ATITUDINI

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

COMPETENȚE SPECIFICE ȘI CONȚINUTURI

11. Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora

Competențe specifice	Conținuturi
----------------------	-------------

12. Elaborarea algoritmilor de rezolvare a problemelor

Competențe specifice	Conținuturi
2.4 Analiza problemei în scopul identificării metodei de programare adecvate pentru rezolvarea problemei 2.5 Aplicarea creativă a metodelor de programare pentru rezolvarea unor probleme intradisciplinare sau interdisciplinare, sau a unor probleme cu aplicabilitate practică 2.6 Analizarea comparativă a eficienței diferitelor metode de rezolvare a aceluiași probleme și alegerea celui mai eficient algoritm de rezolvare a unei probleme	1. Metode de programare <ul style="list-style-type: none">• Metoda de programare <i>Greedy</i> (descrierea generală a metodei, utilitate, aplicații).• Metoda de programare <i>Backtracking</i> (descrierea generală a metodei, utilitate, aplicații)• Metoda de programare <i>Divide et Impera</i> (descrierea generală a metodei, utilitate, aplicații)• Metoda programării dinamice (descrierea generală a metodei, utilitate, aplicații). 2. Analizarea eficienței unui algoritm.

13. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice	Conținuturi
3.6 Declararea corectă a unui pointer de date 3.7 Utilizarea corectă a pointerilor de date 3.8 Utilizarea subprogramelor predefinite de alocare și eliberare dinamică a memoriei	7. Tipuri specifice pentru adresarea zonei de memorie alocate unei variabile (pointeri / referințe). Declarare, operații specifice. 8. Alocarea dinamică a memoriei. (operații și mecanisme specifice)

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 algoritmic;
- activități de dezvoltare a deprinderilor de organizare a informației în diferite structuri de date;
- 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, arborilor 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 algoritmilor fundamentali din teoria grafurilor pe exemple relevante
- 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;
- implementarea structurilor de date alocate dinamic;
- testarea și analizarea comportamentului programelor pentru diferite date de intrare;
- încurajarea discuțiilor purtate între elevi, exprimarea și ascultarea părerilor fiecăruia.

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

Exemple de aplicații recomandate

I. 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 lanț, drum, ciclu sau circuit în graf, identificarea tuturor ciclurilor de lungime 3 într-un graf, verificarea proprietății de graf complet sau graf turneu, etc)
2. Probleme practice, care solicită aplicarea creativă a algoritmilor fundamentali din teoria grafurilor, cum ar fi:
 - Determinarea unei modalități de conectare a unor calculatoare în rețea astfel încât costurile de conectare să fie minime
 - Determinarea unui traseu de lungime minimă între două localități a căror poziție pe hartă este specificată

- Determinarea unei modalități de transmitere a unui mesaj într-o interrețea astfel încât numărul total de servere prin intermediul cărora este transmis mesajul să fie minim.
- Determinarea structurii relaționale a unui grup de persoane

II. Structuri de date alocate dinamic

1. Aplicații simple care să necesite implementarea operațiilor elementare pe liste înlănțuite cum ar fi:
crearea unei liste prin inserări succesive, astfel încât la fiecare pas lista să fie ordonată, inversarea legăturilor într-o listă simplu înlănțuită, numărarea elementelor dintr-o listă cu o anumită proprietate, etc
2. Probleme mai complexe, în care elevii să identifice eficiența utilizării listelor simplu înlănțuite, cum ar fi: reprezentarea grafurilor prin liste de adiacență alocate dinamic, reprezentarea polinoamelor rare în formă condensată și implementarea operațiilor specifice polinoamelor, etc

III. Structuri de date arborescente

1. Aplicații simple care să necesite implementarea operațiilor elementare pe structuri arborescente cum ar fi: parcurgerea unui arbore în scopul identificării tuturor nodurilor cu o anumită proprietate, determinarea înălțimii unui arbore, copierea unui arbore, etc.
2. Probleme mai complexe, în care elevii să identifice eficiența utilizării structurilor de date arborescente, cum ar fi: realizarea eficientă a unui dicționar, sortarea unei secvențe de valori cu ajutorul *heap*-urilor (*heapsort*), optimizarea algoritmului lui *Kruskal* prin organizarea muchiilor ca *heap*.

IV. Metoda de programare Greedy

1. Problema rucsacului în variantă continuă
2. Determinarea arborelui parțial de cost minim (algoritmul lui *Kruskal*, algoritmul lui *Prim*)

V. Metoda de programare Divide et Impera

3. Sortarea eficientă a unei mulțimi de valori aplicând metoda *Divide et Impera* (sortarea rapidă, sortarea prin interclasare)
4. Căutarea eficientă a unui element într-o mulțime ordonată aplicând metoda *Divide et Impera* (căutarea binară)
5. Generarea unor modele fractale

VI. Metoda de programare Backtracking

1. Generarea permutărilor, combinărilor, aranjamentelor, funcțiilor surjective, partițiilor unui număr, partițiilor unei mulțimi
2. Generarea tuturor posibilităților de a ieși dintr-un labirint
3. Generarea tuturor grafurilor parțiale ale unui graf
4. Determinarea tuturor ciclurilor hamiltoniene într-un graf

VII. Metoda programării dinamice

1. Determinarea unui subșir crescător de lungime maximă
2. Înmulțirea optimală a unui șir de matrice.
3. Problema rucsacului în varianta discretă

4. Algoritmul Roy-Floyd de determinare a drumurilor de cost minim între oricare două vârfuri ale unui graf.

VII. Analiza comparativă a rezolvării unei probleme prin diferite metode de programare. De exemplu, problema determinării unui traseu de la vârful unui triunghi către baza acestuia, astfel încât suma elementelor care aparțin traseului să fie minimă; deplasările posibile sunt din elementul curent la unul dintre elementele situate sub el, în stânga sau în dreapta.

Anexa ... la Ordinul ministrului educației și cercetării nr.

..... /

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII

CONSILIUL NAȚIONAL PENTRU CURRICULUM

PROGRAMĂ ȘCOLARĂ PENTRU CICLUL SUPERIOR AL LICEULUI

INFORMATICĂ

CLASA A XII-A

Filiera teoretică, profil real, specializarea: Matematică-informatică

Filiera vocațională, profil militar, specializarea: Matematică-informatică

Aprobat prin Ordinul ministrului

nr. /

București, 2006

NOTA DE PREZENTARE

Prezentul curriculum școlar cuprinde programa școlară pentru disciplina *Informatică*, studiată în clasa a XII-a, la filiera teoretică, profil real, specializarea matematică-informatică, și la filiera vocațională, profil militar, specializarea matematică-informatică.

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 conformitate cu prevederile ordinului ministrului educației și cercetării nr. 5718/ 22.12.2005, cu privire la aprobarea planurilor-cadru de învățământ pentru ciclul superior al liceului, pentru disciplina *Informatică* este prevăzut următorul buget de timp:

- pentru specializarea matematică-informatică: **4 ore/ săptămână**, din care **o oră** pentru activități teoretice și **trei ore** pentru activități practice;
- pentru specializarea matematică-informatică, intensiv informatică: **7 ore/ săptămână**, din care **două ore** pentru activități teoretice și **cinci ore** pentru activități practice.

Programa are următoarele componente:

- ♦ Notă de prezentare
- ♦ Competențe generale
- ♦ Valori și atitudini
- ♦ Competențe specifice și conținuturi
- ♦ Sugestii metodologice

Competențe specifice, definite pentru disciplina *Informatică* la nivelul clasei a XII-a, sunt derivate din competențele generale și reprezintă ansambluri structurate de cunoștințe și deprinderi ce urmează a fi dobândite de către elevi prin învățare, pe durata anului de studiu.

Studiul disciplinei *Informatică* se va desfășura cu întreg colectivul de elevi ai clasei pentru activitățile teoretice, respectiv în laboratorul de informatică pentru activitățile practice. Pentru specializarea matematică-informatică, intensiv informatică, la activitățile practice colectivul de elevi va fi împărțit în două grupe.

Programa are o structură modulară, după cum urmează:

Nr.	Modul	Nr. ore teorie	Nr. ore activități practice	Precizări
1.	Baze de date	1	0	Studiul acestui modul este obligatoriu.
2.	Sisteme de gestiune a bazelor de date A. Modelare date și programare SQL (Oracle) B. Microsoft Visual Fox Pro	0	3	Studiul acestui modul reprezintă o extindere a modulului Baze de date, este opțional, și se poate realiza în una dintre cele două variante specificate (Oracle sau Microsoft Visual Fox Pro), numai în laboratorul de informatică.
3.	Programare orientată pe obiecte și programare vizuală	1 0	2 3	Studiul acestui modul este opțional și se poate realiza în 1 oră de teorie și 2 ore de activități practice sau 3 ore de activități practice.
4	Programare web	1 0	2 3	Studiul acestui modul este opțional și se poate realiza în 1 oră de teorie și 2 ore de activități practice sau 3 ore de activități practice.

Așa cum se deduce din tabelul precedent, studierea modulului 1 (Baze de date) este obligatorie pentru toți elevii de la clasele cu specializarea matematică-informatică. Pentru completarea numărului de ore de studiu, profesorul, de comun acord cu elevii clasei, va alege orice alt modul/combinatie de module, în concordanță cu tabelul precedent.

Studiul modului 2 necesită 3 ore săptămânal și se poate realiza numai în laboratorul de informatică. În cazul selectării variantei A (Modelare date și programare SQL) laboratorul va avea obligatoriu conexiune la Internet pentru a accesa suportul de curs online elaborat de Oracle Academy. Profesorul va opta pentru una dintre cele două variante de sisteme de gestiune a bazelor de date (Oracle sau Microsoft Visual Fox Pro).

Studiul modulelor 3 și 4 se poate realiza într-o oră teorie și două ore activități practice sau 3 ore activități practice. În cazul în care se decide selectarea simultană a modulelor 3 și 4, pentru unul singur dintre acestea se va alege structura cu o oră de teorie.

Variante de studiu pentru specializarea matematică-informatică:

- I. Baze de date (1 oră teorie) + Sisteme de gestiune a bazelor de date (3 ore activități practice).
- II. Baze de date (1 oră teorie) + Programare orientate pe obiecte și programare vizuală (3 ore activități practice).
- III. Baze de date (1 oră teorie) + Programare web (3 ore activități practice).

Variante de studiu pentru specializarea matematică-informatică, intensiv informatică:

- I. Baze de date (1 oră teorie) + Sisteme de gestiune a bazelor de date (3 ore activități practice) + Programare orientate pe obiecte și programare vizuală (1 oră teorie + 2 ore activități practice).
- II. Baze de date (1 oră teorie) + Programare web (1 oră teorie și 2 ore activități practice) + Programare orientată pe obiecte și programare vizuală (3 ore activități practice).
- III. Baze de date (1 oră teorie) + Programare orientată pe obiecte și programare vizuală (1 oră teorie și 2 ore activități practice) + Programare web (3 ore activități practice).
- IV. Baze de date (1 oră teorie) + Sisteme de gestiune a bazelor de date (3 ore activități practice) + Programare web (1 oră teorie + 2 ore activități practice).

Parcursul modulelor selectate se poate realiza secvențial sau în paralel.

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.

COMPETENȚE GENERALE

Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora
Elaborarea algoritmilor de rezolvare a problemelor
Implementarea algoritmilor într-un limbaj de programare

VALORI ȘI ATITUDINI

- ♦ Exprimarea unui mod de gândire creativ, în structurarea și rezolvarea problemelor.
- ♦ Conștientizarea impactului social, economic și moral al informaticii.
- ♦ Formarea obișnuințelor de a recurge la concepte și metode informatice de tip algoritmic specifice în abordarea unei varietăți de probleme.
- ♦ Manifestarea unor atitudini favorabile față de știință și de cunoaștere în general.
- ♦ 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
1.1. Analizarea unei probleme în scopul identificării și clasificării datelor necesare	Modelul conceptual al problemei (entități, proprietăți, comportamente) Tipuri de date
1.2. Identificarea relațiilor dintre date	Relații între entități
1.3. Identificarea modalităților adecvate de structurare a datelor care intervin într-o problemă	Modele de organizare a datelor. Structuri de date.
1.4. Utilizarea funcțiilor specifice de prelucrare a datelor structurate	Operații elementare de prelucrare a datelor

2. Elaborarea algoritmilor de rezolvare a problemelor

Competențe specifice	Conținuturi
2.1. Identificarea tehnicilor de programare adecvate rezolvării unei probleme și aplicarea creativă a acestora	Tehnici de prelucrare a datelor. Instrucțiuni specifice limbajului de programare.
2.2. Elaborarea strategiei de rezolvare a unei probleme	Structura unei aplicații. Modularizarea aplicației.
2.3. Analizarea comparativă a eficienței diferitelor tehnici de rezolvare a problemei respective și alegerea celei mai eficiente variante	Criterii de eficiență a aplicațiilor.

3. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice	Conținuturi
3.1. Utilizarea instrumentelor de dezvoltare a unei aplicații	Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice).
3.2. Elaborarea și realizarea unei aplicații, folosind un mediu de programare specific	Etape în dezvoltarea aplicațiilor.
3.3. Prezentarea unei aplicații	Reguli elementare pentru crearea și susținerea unei prezentări publice.

CONȚINUTURI DETALIAȚE

Modulul 1: BAZE DE DATE

Modelul conceptual al unei probleme de gestiune

- Entități și instanțe
- Atribute
- Identificator unic
- Relații între entități (one-to-one, one-to-many, many-to-many)
- Rezolvarea relațiilor many-to-many
- Normalizarea datelor: prima forma normală, a doua forma normală, a treia formă normală

Tabele

- Crearea structurii tabelor (tipuri de date, structură, câmpuri/coloane)
- Conținutul unei tabele (linii/înregistrări)
- Operații specifice prelucrării tabelor (adăugare, modificare, ștergere, sortare, căutare, vizualizare, calcule statistice)

Baze de date

- Modele de baze de date (modelul relațional, modelul rețea, modelul ierarhic)
- Relaționare, cheie primară, chei externe
- Reguli de integritate
- Programe de validare, de acțiune
- Operații specifice prelucrării bazelor de date (interogări, rapoarte)

Dezvoltarea profesională în domeniul IT

- Identificarea aptitudinilor pentru anumite tipuri de activități
- Crearea unui CV și reguli de susținere a unui interviu
- 10 reguli în susținerea unei prelegeri
- Principii de lucru în echipă

Modulul 2: SISTEME DE GESTIUNE A BAZELOR DE DATE

VARIANTA A. MODELARE DATE ȘI PROGRAMARE SQL (ORACLE)

Modelul conceptual al aplicațiilor

- Convenții pentru realizarea diagramelor ERD
- Tipuri și subtipuri
- Transferabilitate
- Relații ierarhice, relații recursive
- Arce

Introducere în SQL; structura comenzilor SQL; operații elementare asupra datelor

- Selecție, proiecție
- Expresii, funcții
- Gruparea datelor
- Sortarea datelor
- Interogări simple
- Relaționarea tabelor
- Interogări multiple
- Crearea și modificarea structurii tabelor
- Inserarea, modificarea, ștergerea datelor în tabele
- Constrângeri
- Crearea și utilizarea secvențelor
- Indecși și sinonime
- Crearea și revocarea privilegiilor
- Gestionarea tranzacțiilor

Realizarea unui proiect folosind Oracle SQL

- Lucrul în echipă pentru elaborarea unui proiect după un plan dat
- Realizarea modelului conceptual, harta relațiilor
- Construirea bazei de date și implementarea folosind Application Builder

VARIANTA B. MICROSOFT VISUAL FoxPro

Mediul de programare Microsoft Visual FoxPro

Facilități oferite în lucrul cu baze de date

Moduri de lucru (asistat, comandă, program)

Obiecte vizuale (ferestre, butoane, liste etc.), proprietăți (atribute, condiții, evenimente)

Tipuri de date

Tabele simple

Stabilirea structurii unei tabele simple (CREATE, CREATE TABLE)

Afișarea, modificarea structurii unei tabele (DISPLAY/LIST/MODIFY STRUCTURE)

Introducerea, modificarea, ștergerea datelor în/din tabele (APPEND, REPLACE, DELETE, RECALL, PACK, CHANGE, EDIT, BROWSE)

Afișarea datelor din tabele (LIST, DISPLAY)

Poziționarea și căutarea în tabele (GO, SKIP, LOCATE, CONTINUE)

Sortarea înregistrărilor unei tabele (SORT)

Lucrul simultan cu mai multe tabele. Zone de lucru (SELECT)

Comenzi de calcul (CALCULATE)

Editarea, compilarea și rularea unui program (MODIFY COMMAND, DO)

Instrucțiuni

Instrucțiuni simple de citire/scriere (INPUT, ACCEPT, ?, ??, ???)

Instrucțiuni condiționale (IF, DO CASE)

Instrucțiuni repetitive (DO WHILE, FOR, SCAN, LOOP, EXIT)

Proceduri și funcții

Funcții standard

Proceduri și funcții utilizator (definire, apel, transmiterea informațiilor prin parametri)

Proiecte (Project Manager)

Baze de date

Operații asupra tabelelor incluse în baza de date

Filtre

Indeși în Visual FoxPro (INDEX, SET ORDER, DELETE TAG)

Relații între tabele (SET RELATION, SET SKIP, JOIN)

Limbajul SQL

Comenzi SQL (SELECT, INSERT, UPDATE, DELETE, Query Designer)

Formulare (Form Designer)

Creare și caracteristici

Utilizarea controalelor (butoane de comandă, butoane radio, butoane de tip casetă cu incrementare, casete de validare, casete de text, casete de editare, imagini, liste, etc.),

Fixarea proprietăților și stabilirea răspunsului la evenimente, stabilirea ierarhizării obiectelor din formular

Rapoarte (Report Designer)

Etichete (Label Designer)

Meniuri (Menu Builder)

Proiect de gestiune a datelor specifice unui domeniu de interes practic.

Notă: Setul de comenzi specificat în programă este minimal.

Modulul 3. PROGRAMAREA ORIENTATĂ PE OBIECTE ȘI PROGRAMARE VIZUALĂ

► PROGRAMAREA ORIENTATĂ PE OBIECTE

- **Principiile programării orientate pe obiecte**
- **Structura unei aplicații orientată pe obiecte**
- **Clase și obiecte**
 - Definirea claselor și a obiectelor
 - Utilizarea claselor și a obiectelor
 - Specificatori de acces la membrii unei clase
 - Supraîncărcarea funcțiilor
 - Crearea și distrugerea obiectelor; constructori și destructor
 - Supraîncărcarea operatorilor
 - Funcții prietene (*friend*)
 - Membri statici ai unei clase
 - Modificatorul `const`
- **Clase și funcții șablon (template)**
 - Necesitatea utilizării claselor/funcțiilor șablon
 - Declararea unei clase șablon
 - Utilizarea unei clase șablon
 - Biblioteca de clase șablon STL (Standard Template Library) - prezentare generală
- **Derivarea claselor**
 - Definirea noțiunii de moștenire simplă și moștenire multiplă
 - Declararea unei clase derivate
 - Drepturi de acces în clase derivate
 - Constructorii și destructorii în clase derivate
- **Tratarea erorilor**
 - Ce este o eroare
 - Cum se generează
 - Tratarea/propagarea erorilor
 - Ierarhia claselor de erori
- **Polimorfism**
 - Funcții virtuale
 - Clase abstracte și funcții virtuale pure

► PROGRAMAREA VIZUALĂ

- **Concepte de bază ale programării vizuale**
- **Prezentarea unui mediu de programare vizual (Microsoft Visual C#, Delphi, Microsoft Visual Basic etc.).**
 - Operații și unelte specifice ale acestuia
- **Elementele POO în context vizual**
- **Construirea interfeței utilizator**
 - Ferestre: tipuri, utilizare, organizare, proprietăți, evenimente
 - Controale: tipuri, utilizare, organizare, proprietăți, evenimente
 - Meniuri
 - Bare de unelte
 - Obiecte grafice
- **Accesarea și prelucrarea datelor**
 - Citiri și scrieri utilizând fișiere și dispozitive standard
 - Crearea unei baze de date, conectarea și deconectarea la o bază de date
 - Popularea bazei de date
 - Manipularea datelor dintr-o bază de date
 - Interogarea datelor dintr-o bază de date

► DEZVOLTAREA ȘI PREZENTAREA UNEI APLICAȚII ÎN MEDIU VIZUAL

Modulul 4. PROGRAMARE WEB

- **Principii generale ale proiectării interfețelor Web**
 - Etapele procesului de dezvoltare a unei aplicații Web
 - Aspecte generale ale proiectării interfețelor Web
 - Realizarea interfețelor Web utilizând limbajul de marcare HTML (elemente avansate):
tabele, formulare, cadre, layer-e
 - Foi de stiluri (CSS)
- **Modelul client-server. Protocoale de comunicație**
- **Mediul de lucru (server web - Apache, IIS etc., instrumente de dezvoltare a aplicațiilor - PHPdev etc.)**
- **Prezentarea unui limbaj de scripting server-side (PHP, ASP etc.)**
 - Elemente de bază ale limbajului
 - Instrucțiuni
 - Funcții
 - Structuri de date
- **Interacțiunea cu baze de date Web (MySQL, SQL Server etc.)**
 - Aplicații pentru definirea și gestionarea unei baze de date
 - Conectare/deconectare la baza de date
 - Transmiterea interogărilor SQL către baza de date. Preluarea și prelucrarea datelor returnate de interogările SQL
- **Proiectarea și realizarea unei aplicații Web**

SUGESTII METODOLOGICE

Predarea-învățarea 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 pentru aplicarea corectă a programei se sugerează următoarele activități de învățare:

- exerciții de descriere a metodei de rezolvare a unei probleme din perspectiva structurării datelor;
- testarea și analizarea comportamentului aplicațiilor pentru diferite date de intrare;
- discuții privind validitatea datelor;
- proiectarea etapelor unei activități;
- găsirea unor date reprezentative pentru cazuri generale, respectiv descoperirea cazurilor particulare;
- dezbateri pe tema fixării rolurilor în echipă în funcție de interesele și aptitudinile individuale;
- formularea unor probleme care să poată fi realizate în grupuri de elevi pe baza unor discuții preliminare și analiza problemei;
- discuții de abordare a problemelor care apar pe parcursul desfășurării activităților;
- verificarea înțelegerii rezolvării unei probleme în ansamblul ei de către toți membrii grupului;
- evidențierea necesității realizării corecte a unei aplicații;
- prezentarea și dezbaterile aplicațiilor realizate;
- organizarea de discuții între elevi, exprimarea și ascultarea părerilor fiecăruia;
- 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;
- evidențierea importanței realizării unor produse program fiabile, cu interfață prietenoasă;
- pentru înțelegerea principiilor programării orientată pe obiecte se va prezenta o aplicație din viața reală, familiară elevului;
- pentru a ilustra principiile programării orientată pe obiecte și modul de structurare a unei aplicații POO se va prezenta și analiza o aplicație gata implementată; se va realiza o analiză comparativă între abordarea POO în raport cu abordarea specifică programării structurate;
- conceptele fundamentale ale POO vor fi prezentate diferențiat, în funcție de limbajul de programare studiat de elev; acolo unde este cazul, se pot evidenția diferențele specifice altor limbaje de programare;

- pentru ca elevii să înțeleagă diferențele specifice abordării POO, se va alege una dintre aplicațiile familiare elevilor (de exemplu: numere complexe, numere raționale, polinoame, liste, etc.), care va fi analizată din perspectiva POO, apoi se vor implementa clasele necesare și se va realiza o aplicație în care vor fi utilizate clasele create;
- se va analiza un exemplu de ierarhie de clase gata implementată, pentru a înțelege mecanismele care guvernează derivarea claselor;
- fiecare elev va alege o problemă concretă, va defini clasele/ierarhiile de clase necesare și va crea o aplicație în care să fie utilizate clasele/ierarhiile de clase create; în acest scop se poate folosi un mediu vizual; elevii vor prezenta colegilor aplicațiile create; profesorul va stimula discuțiile critice și creative pe marginea aplicațiilor elaborate de elevi;
- conceptele fundamentale ale programării vizuale se vor prezenta și analiza pe o aplicație gata implementată; se vor pune în evidență elementele specifice: obiecte vizuale, proprietățile și evenimentele legate de acestea;
- Se va da importanță analizei și proiectării interfeței corespunzătoare unei aplicații vizuale.
- se va pune accent pe noțiunile legate de interfața cu utilizatorul ca parte importantă a aplicației. interfața trebuie să satisfacă toate cerințele proiectului. se va accentua importanța designului în crearea unei interfețe;
- în elaborarea aplicațiilor se va pune accent pe documentarea aplicației.

Pentru modulul 2, *Sisteme de gestiune a bazelor de date*, varianta A (Modelare date și programare SQL) se utilizează cursul online Oracle Academy.

Programa școlară de *Informatică* pentru clasa a XII-a a fost realizată de grupul de lucru pentru curriculum din cadrul comisiei naționale de specialitate și a primit avizul acesteia.

I. Grupul de lucru pentru curriculum:

prof. gr. I. Rodica Pinte, Liceul Teoretic "Grigore Moisil" București
 prof. gr. I Emil Onea, I.S.J. Vrancea
 prof. gr. I. Emanuela Cerchez, Liceul de Informatică "Grigore Moisil" Iași
 șef lucrări Stelian Ciurea, Universitatea "Lucian Blaga" Sibiu
 prof. gr. I. Maria Niță, Colegiul Național "Emanuil Gojdu" Oradea
 prof. gr. I. Adrian Niță, Colegiul Național "Emanuil Gojdu" Oradea
 prof. gr. I. Marinel Șerban, Liceul de Informatică "Grigore Moisil" Iași
 prof. gr. I. Roxana Tîmplaru, I.S.J. Dolj
 prof. gr. II. Alin Burța, Colegiul Național "B. P. Hasdeu" Buzău
 prof. gr. I. Dan Grigoriu, Colegiul Național de Informatică "Tudor Vianu" București

II. Comisia națională de informatică:

Nr. crt.	Nume și Prenume	Titlu științific/ grad didactic	Instituția de învățământ	Județul
1.	Nușa Dumitriu-Lupan	profesor gr. I	M.Ed.C. București	București
2.	Adrian Atanasiu	prof. univ. dr.	Universitatea București	București
3.	Brândușa Bogdan	profesor gr. I	I.S.M. București	București
4.	Alin Burța	profesor gr. II	Colegiul Național "B. P. Hasdeu", Buzău	Buzău
5.	Emanuela Cerchez	profesor gr. I	Liceul de Informatică "Grigore Moisil" Iași	Iași
6.	Stelian Ciurea	șef lucrări drd.	Universitatea „Lucian Blaga”, Sibiu	Sibiu
7.	Ovidiu Domșa	asist. univ. drd.	Universitatea "1 Decembrie 1918" Alba-Iulia	Alba
8.	Dan Grigoriu	profesor gr. I	Colegiul Național de Informatică „Tudor Vianu” București	București
9.	Sanda Junea	profesor gr. I	Liceul "Grigore Moisil" Timișoara	Timiș
10.	Nistor Eugen Moț	profesor gr. I	Colegiul Național "Nicolae Bălcescu" Brăila	Brăila
11.	Maria Niță	profesor gr. I	Colegiul Național „Emanuil Gojdu” Oradea	Bihor
12.	Adrian Niță	profesor gr. I drd.	Colegiul Național „Emanuil Gojdu” Oradea	Bihor
13.	Emil Onea	profesor gr. I	I.S.J. Vrancea	Vrancea
14.	Rodica Pinte	profesor gr. I	Liceul Teoretic "Grigore Moisil" București	București
15.	Doru Popescu Anastasiu	profesor gr. I drd	Colegiul Național "Radu Greceanu" Slatina	Olt
16.	Vasile Roman	profesor gr. I	Școala nr. 7 "Sfânta Maria" Timișoara	Timiș
17.	Marinel Șerban	profesor gr. I	Liceul de Informatică "Grigore Moisil" Iași	Iași
18.	Roxana Tîmplaru	profesor gr. I	I.S.J. Dolj	Dolj
19.	Giorgie-Daniel Vlad	profesor gr. I	I.S.J. Suceava	Suceava

1

MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI TINERETULUI
CONSILIUL NAȚIONAL PENTRU CURRICULUM
CURRICULUM ȘCOLAR

pentru

INFORMATICA

Profil real

Specializarea:

Matematică-informatică, intensiv informatică

Clasa a IX-a

București

2004

2

Notă de prezentare

Prezentul curriculum școlar cuprinde programa pentru disciplina **Informatică**, studiată în filiera

teoretică, la profilul real, specializarea matematică-informatică, intensiv informatică, prevăzută

săptămânal cu două ore pentru activități teoretice și două ore pentru activități practice, în conformitate cu art. 9. alin c) din OMECT 5723/23.12.2003.

Programa are următoarele componente:

- Notă de prezentare
- Competențe generale
- Competențe specifice și conținuturi
- Valori și atitudini
- Sugestii metodologice.

În elaborarea programei de față au fost luate în considerație atât cercetările în domeniul curricular, tendințe pe plan internațional, cât și opiniile unor profesori cu o bogată experiență didactică.

Ținând cont de noutatea formei de prezentare, sunt necesare următoarele completări:

– **Competențele generale** sunt ansambluri structurate de cunoștințe și deprinderi; ele se definesc pe obiect de studiu și se formează pe durata unui ciclu curricular.

– **Competențele specifice** se definesc pe obiect de studiu și se formează pe durata unui an de

studiu. Ele sunt derivate din competențele generale.

Studiul disciplinei Informatică se va desfășura cu întreg colectivul de elevi ai clasei pentru

activităților teoretice și cu colectivul de elevi organizat pe grupe, obligatoriu în laboratorul de

informatică pentru activitățile practice.

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.

3

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**

4

INFORMATICĂ

Clasa a IX-a

A. COMPETENȚE SPECIFICE ȘI CONȚINUTURI

1. Identificarea conexiunilor dintre informatică și societate.

Competențe specifice Conținuturi

–Identificarea aplicațiilor informaticii în viața socială

–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

–Descompunerea rezolvării unei probleme în pași

–Identificarea tipurilor de date necesare pentru rezolvarea unei probleme (de intrare, de ieșire, de manevră).

–Descrierea coerentă a unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire.

Etapele rezolvării problemelor. Exemple

Noțiunea de algoritm. Caracteristici. Exemple.

Date cu care lucrează algoritmii (constante, variabile, expresii).

Operații asupra datelor (aritmetice, logice, relaționale).

3. Elaborarea algoritmilor de rezolvare a problemelor

Competențe specifice Conținuturi

–Analizarea enunțului unei probleme:
identificarea datele de intrare și a datele
de ieșire (cu specificarea tipul datelor
și a relațiilor existente între date) și
stabilirea pașilor de rezolvare a
problemei.

–Reprezentarea algoritmilor în
pseudocod

–Respectarea principiilor programării
structurate în procesul de elaborare a
algoritmilor

Reprezentarea algoritmilor în pseudocod.

**Principiile programării structurate. Structuri de
bază:** structura liniară, structura alternativă, structura
repetitivă

Algoritmi elementari

1. Prelucrarea numerelor :

– 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)

2. Prelucrarea unor secvențe de valori

– 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)

5

4. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice Conținuturi

–Transcrierea algoritmilor din limbaj
pseudocod în limbaj de programare

–Identificarea necesității structurării
datelor în tablouri

–Prelucrarea datelor structurate

–Utilizarea fișierelor text pentru introducerea
datelor și extragerea

rezultatelor

–Utilizarea unui mediu de programare

(pentru limbajul Pascal sau pentru limbajul C/C++)

Elementele de bază ale limbajului de programare

Noțiuni introductive

- Structura programelor
- Vocabularul limbajului
- Tipuri simple de date (standard)
- Constante, variabile, expresii
- Citirea/scrierea datelor

Structuri de control

- Structura liniară
- Structura alternativă
- Structuri repetitive

Tipuri structurate de date. Tipul tablou

- Tablouri unidimensionale
- Tablouri bidimensionale

Algoritmi fundamentali de prelucrare a datelor structurate în tablouri

- căutare secvențială, căutare binară
- sortare
- interclasare
- parcurgerea tablourilor bidimensionale pe linii/coloane.

Fișiere text. Definiție, operații specifice

Mediul limbajului de programare studiat

- Prezentare generală
- Editarea programelor sursă
- Compilare, rulare, depanare

6. Aplicarea algoritmilor fundamentali în prelucrarea datelor

Competențe specifice Conținuturi

- Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării
- Alegerea celui mai eficient algoritm de rezolvare a unei probleme

Aplicații interdisciplinare (specifice profilului).

Exemple:

- 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.

Analizarea unui algoritm din punctul de vedere al numărului de operații executate.

6

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

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 structurilor de bază în pseudocod;

– exersarea scrierii unor algoritmi simpli, folosind structuri lineare, 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/modelarea 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.

1

MINISTERUL EDUCAȚIEI, CERCETĂRII ȘI TINERETULUI
CONSILIUL NAȚIONAL PENTRU CURRICULUM
CURRICULUM ȘCOLAR

pentru

INFORMATICA

Profil real

Specializarea:

Matematică-informatică

Științe ale naturii

Clasa a IX-a

București

2004

2

Notă de prezentare

Prezentul curriculum școlar cuprinde programa pentru disciplina **Informatică**, studiată în filiera

teoretică, la profilul real, specializările matematică-informatică și științele naturii, prevăzută cu o

oră pe săptămână în componenta curriculum diferențiat.

Programa are următoarele componente:

- Notă de prezentare
- Competențe generale
- Competențe specifice și conținuturi
- Valori și atitudini
- Sugestii metodologice.

În elaborarea programei de față au fost luate în considerație atât cercetările în domeniul curricular, tendințe pe plan internațional, cât și opiniile unor profesori cu o bogată experiență didactică.

Ținând cont de noutatea formei de prezentare, sunt necesare următoarele completări:

– **Competențele generale** sunt ansambluri structurate de cunoștințe și deprinderi; ele se definesc pe obiect de studiu și se formează pe durata unui ciclu curricular.

– **Competențele specifice** se definesc pe obiect de studiu și se formează pe durata unui an de

studiu. Ele sunt derivate din competențele generale.

Studiul disciplinei Informatică se va desfășura cu întreg colectivul de elevi ai clasei.

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.

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

3

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**

4

INFORMATICĂ

Clasa a IX-a

A. COMPETENȚE SPECIFICE ȘI CONȚINUTURI

1. Identificarea conexiunilor dintre informatică și societate.

Competențe specifice Conținuturi

– Identificarea aplicațiilor informaticii în viața socială

– Recunoașterea situațiilor în care este necesară prelucrarea algoritmică a informațiilor.

Definirea informaticii ca știință

Rolul informaticii în societate

Studii de caz al unor situații sociale, în abordare informatizată

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

Competențe specifice Conținuturi

– Descompunerea rezolvării unei probleme în pași

– Identificarea tipurilor de date necesare pentru rezolvarea unei probleme (de intrare, de ieșire, de manevră).

– Descrierea coerentă a unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire.

Etapele rezolvării problemelor. Exemple

Noțiunea de algoritm. Caracteristici. Exemple.

Date cu care lucrează algoritmii (constante, variabile, expresii).

Operații asupra datelor (aritmetice, logice, relaționale).

3. Elaborarea algoritmilor de rezolvare a problemelor

Competențe specifice Conținuturi

– Analizarea enunțului unei probleme:

identificarea datele de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei.

- Reprezentarea algoritmilor în pseudocod
- Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor

Reprezentarea algoritmilor în pseudocod.

Principiile programării structurate. Structuri de bază:

- structura liniară
- structura alternativă
- structura repetitivă

Algoritmi elementari

1. Prelucrarea numerelor :

- 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)

2. Prelucrarea unor secvențe de valori

- 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)

5

- generarea șirurilor recurente (de exemplu: șirul Fibonacci)

4. Aplicarea algoritmilor fundamentali în prelucrarea datelor

Competențe specifice Conținuturi

- Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării

- Alegerea celui mai eficient algoritm de rezolvare a unei probleme

Aplicații interdisciplinare (specifice profilului).

Exemple:

- 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, etc)

–Determinarea punctului de intersecție a două mobile în mișcare rectilinie și uniformă

–Determinarea masei moleculare a unui compus chimic.

Analizarea unui algoritm din punctul de vedere al numărului de operații executate.

5. Implementarea algoritmilor într-un limbaj de programare

Competențe specifice Conținuturi

–Identificarea într-un program a structurilor de control învățate

Exemplificări de modalități de implementare a algoritmilor studiați

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

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 activități complexe în

funcție de scopul propus;

– explicarea conceptului de algoritm și a caracteristicilor algoritmilor;

– explicarea diferenței existente între informații care se materializează în date concrete și cele

care determină calea de rezolvare a unei probleme;

– descrierea unui algoritm în limbaj natural;

6

– prezentarea obiectelor cu care operează algoritmii (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 vocabularului pseudocodului și a structurilor de bază: structura liniară, alternativă și

repetitivă;

- exersarea scrierii unor algoritmi simpli, folosind structuri lineare, alternative și repetitive;

- identificarea diferitelor situații în care alegerea unui tip de 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;

- încurajarea discuțiilor purtate între elevi, exprimarea și ascultarea părerilor fiecăruia.

<http://www.edu.ro/pr102005.htm>
Anexa nr. 2 la ordinul ministrului educației și cercetării nr. 4598 /
31.08.2004

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII

CONSILIUL NAȚIONAL PENTRU CURRICULUM