

MINISTERUL EDUCAȚIEI
CENTRUL NAȚIONAL DE POLITICI ȘI EVALUARE ÎN EDUCAȚIE

REPERE METODOLOGICE

PENTRU APLICAREA CURRICULUMULUI LA
CLASA a XII-a
ÎN ANUL ȘCOLAR 2024-2025

**Orientarea procesului educativ la disciplina/disciplinele/modulele de studiu,
în vederea atingerii de ținte și obiective stabilite prin
documente de politică educațională/metodologii**

**Valorizarea de oportunități asigurate prin
proiectele și programele Ministerului Educației**

DISCIPLINA INFORMATICĂ

București 2024

CUPRINS

I. PREMISE PENTRU APLICAREA CURRICULUMULUI LA CLASA A XII-A ÎN ANUL ȘCOLAR 2024 - 2025	2
I.1. Specificul clasei a XII-a privind disciplina informatică	2
I.2. Situația disciplinei în ansamblul curriculumului național	2
I.2.1. Programele școlare pentru disciplina informatică	2
I.2.2. Examene la absolvirea clasei a XII-a	5
<i>Examenul de atestare a competențelor profesionale</i>	5
Examenul național de bacalaureat la disciplina informatică	6
I.2.3. Manuale pentru disciplina informatică	8
I.3. Structura anului școlar 2024-2025	9
I.3.1. Exemple orientative de planificări calendaristice anuale	10
I.4. Orientări privind proiectarea didactică	40
II. ORIENTAREA PROCESULUI DE PREDARE-ÎNVĂȚARE-EVALUARE LA DISCIPLINA INFORMATICĂ, AVÂND CA REFERENȚIAL PROFILUL DE FORMARE AL ABSOLVENTULUI ȘI PENTRU PREGĂTIREA ELEVILOR ÎN VEDEREA SUSȚINERII EXAMENULUI NAȚIONAL DE BACALAUREAT ȘI A EXAMENELOR DE CERTIFICARE A COMPETENȚELOR PROFESIONALE	41
II.1. Exemplu de activitate de învățare care să susțină formarea de competențe specifice	41
Modulul de programare vizuală	41
II.2. Exemple de teme pentru recapitulare/sinteză, în contextul susținerii, la finalul anului școlar, a examenelor specifice	47
II.2.1. Teme pentru recapitulare - divizibilitate	47
II.2.2. Teme pentru recapitulare – șiruri de caractere	54
II.2.3. Teme pentru recapitulare – Metoda backtracking	59
II.2.4. Teme pentru recapitulare – Grafuri	62
II.3. Evaluarea în mediul on-line a performanțelor școlare	65
II.3.1. Recomandări privind evaluarea în mediul on-line a performanțelor școlare și a competențelor școlare	65
II.3.2. Exemple de platforme/aplicații care sprijină elaborarea și administrarea de instrumente de evaluare	66
II.4. Abordări multi-, inter- și transdisciplinare	68
III. VALORIZAREA DE OPORTUNITĂȚI ASIGURATE PRIN PROIECTELE ȘI PROGRAMELE MINISTERULUI EDUCAȚIEI	72
III.1. Discipline opționale, din oferta națională, care completează curriculumul obligatoriu în domeniul disciplinei/modulului de studiu, în relație directă/indirectă cu acesta	73
III.2. Ghiduri elaborate în cadrul unor proiecte	74
Ghidul profesorului din România	74
BIBLIOGRAFIE	76
RESURSE WEB	78
COLECTIV DE AUTORI	79

I. PREMISE PENTRU APLICAREA CURRICULUMULUI LA CLASA A XII-A ÎN ANUL ȘCOLAR 2024 - 2025

I.1. Specificul clasei a XII-a privind disciplina informatică

Disciplina informatică se studiază în clasa a XII-a la filiera teoretică, profil real, specializările matematică-informatică și matematică-informatică, intensiv informatică, respectiv la filiera vocațională, profil militar, specializarea matematică-informatică.

I.2. Situația disciplinei în ansamblul curriculumului național

I.2.1. Programele școlare pentru disciplina informatică

Organizarea activității didactice la disciplina informatică la clasa a XII-a, la specializarea matematică informatică, presupune o oră desfășurată în sala de clasă, în care sunt prezentate noțiuni teoretice, și trei ore desfășurate în laboratorul de informatică, în care sunt realizate cu preponderență activități practice. La specializarea matematică-informatică, intensiv informatică, disciplinei i se alocă trei ore desfășurate în sala de clasă, în care sunt prezentate noțiuni teoretice, o oră desfășurată cu întreaga clasă în laboratorul de informatică și alte trei ore desfășurate în laboratorul de informatică, cu clasa organizată pe grupe, în care sunt realizate cu preponderență activități practice.

Programa școlară în vigoare pentru disciplina informatică - clasa a XII-a, filiera teoretică, profil real, specializările matematică-informatică, matematică-informatică intensiv informatică și filiera vocațională, profil militar, specializarea matematică-informatică este aprobată prin *OMECI nr. 5099/09.09.2009*.

Sunt propuse mai multe module, dintre care primul este obligatoriu, iar celelalte sunt opționale, alese astfel încât numărul total de ore alocate disciplinei, respectiv reperelor teoretice/activităților practice să fie conform alocării precizate mai sus.

Nr.	Modul	Nr. ore teorie	Nr. ore activități practice	Precizări
.	Baze de date	1	0	Studierea modului este obligatorie și are o alocare orară săptămânală de 1 oră.
.	Sisteme de gestiune a bazelor de date. A.Modelarea datelor și programare SQL (Oracle, Microsoft SQL Server, MySQL) B.Programare Visual FoxPro (Microsoft)	0	3	Modulul reprezintă o extindere a modulului Baze de date și are ca suport unul dintre sistemele de gestiune a bazelor de date următoare: Oracle, Microsoft SQL Server, MySQL, Visual FoxPro. Studierea modulului este opțională și are o alocare orară săptămânală de 3 ore de activități practice
.	Programare vizuală	1 0	2 3	Studierea modulului este opțională și are o alocare orară săptămânală de 1 oră de teorie și 2 ore de activități practice sau de 3 ore de activități practice.
.	Programare web	1 0	2 3	Studierea modulului este opțională și are o alocare orară săptămânală de 1 oră de teorie și 2 ore de activități practice sau de 3 ore de activități practice.
.	Programarea procedurală a bazelor de date (PL/SQL, Transact-SQL, MySQL).	1	2	Modulul reprezintă o extindere a modulului Sisteme de gestiune a bazelor de date, varianta A, și se poate realiza având în vedere unul dintre cele trei variante de limbaje: PL/SQL, Transact-SQL, MySQL. Studierea modulului este opțională și se poate realiza cu o alocare orară săptămânală de 1 oră de teorie și 2 ore de activități practice.

1. Pentru completarea numărului de ore alocate săptămânal disciplinei, se va alege modulul obligatoriu 1, apoi orice alt modul/combinatie de module, în funcție de numărul de ore disponibile și în concordanță cu tabelul de mai sus.
2. Pentru modulul 2, în cazul selectării variantei A (Modelare date și programare SQL) se recomandă ca laboratorul să fie conectat la Internet, pentru a accesa eventualele suporturi de curs on-line.
3. Pentru specializarea matematică-informatică, intensiv informatică, dacă se optează pentru studierea atât a modulului 3, cât și a modulului 4, pentru unul dintre acestea se va alege o alocare orară săptămânală de 1 oră de teorie și 2 ore de activități practice, iar pentru celălalt de 3 ore de activități practice.
4. Modulul 5 poate fi studiat doar în continuarea modulului 2.

Variante de studiu pentru specializarea matematică-informatică:

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

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

- I. Baze de date (1 oră de teorie) + Sisteme de gestiune a bazelor de date (3 ore de activități practice) + Programarea procedurală a bazelor de date (1 oră de teorie + 2 ore de activități practice).
- II. Baze de date (1 oră de teorie) + Sisteme de gestiune a bazelor de date (3 ore de activități practice) + Programare vizuală (1 oră de teorie + 2 ore de activități practice).
- III. Baze de date (1 oră de teorie) + Programare web (1 oră de teorie și 2 ore de activități practice) + Programare vizuală (3 ore de activități practice).
- IV. Baze de date (1 oră de teorie) + Programare vizuală (1 oră de teorie și 2 ore de activități practice) + Programare web (3 ore de activități practice).
- V. Baze de date (1 oră de teorie) + Sisteme de gestiune a bazelor de date (3 ore de activități practice) + Programare web (1 oră de teorie + 2 ore de activități practice).

I.2.2. Examene la absolvirea clasei a XII-a

Examenul de atestare a competențelor profesionale

Atestatul este un act de studiu prin care se certifică pregătirea de specialitate/profesională dobândită de absolvenții de liceu, cursuri de zi. El se eliberează absolvenților de la clasele de matematică-informatică și matematică-informatică, intensiv informatică, ce au promovat proba de specialitate prevăzută de Metodologia de organizare și desfășurare a examenului de atestare a competențelor profesionale.

Competențele certificate în urma promovării examenului de atestare a competențelor profesionale sunt prevăzute în *Metodologia de organizare și desfășurare a examenului de atestare a competențelor profesionale a absolvenților claselor de matematică-informatică și matematică-informatică intensiv informatică* în vigoare și sunt precizate mai jos:

- Pentru absolvenții claselor de matematică-informatică, intensiv informatică:
 - realizarea design-ului și structurii produselor software necesare implementării de: sisteme software, aplicații software, baze de date, pagini web;
 - particularizarea, configurarea și modificarea aplicațiilor software, în scopul adaptării sistemelor informaționale ale clientului.
- Pentru absolvenții claselor de matematică-informatică:
 - realizarea managementului site-urilor web, sistemelor de operare a calculatoarelor;
 - furnizarea facilităților de procesare a datelor, inclusiv a serviciilor suport.

Proba de specialitate constă în:

- o probă practică: realizarea practică pe calculator a cerințelor din biletul extras în ziua probei.
Biletul cuprinde trei cerințe:
 - privind sistemele de gestiune a bazelor de date;
 - privind scrierea unui algoritm în limbaj de programare;
 - privind sisteme de operare sau abilități de tehnoredactare, calcul tabelar sau prezentări electronice.
- proiect: prezentarea și motivarea teoretică a unui proiect (produs software) realizat în timpul orelor de laborator al ultimului an de studiu; produsele pot fi realizate și în echipă (2 - 3 elevi) în funcție de complexitatea proiectului.

Atestatul, obținut în cazul promovării examenului de atestare a competențelor profesionale, se eliberează de secretariatul unității de învățământ în care a fost susținut examenul, după

promovarea de către candidat a clasei a XII-a, iar în caz contrar examenul de atestare susținut și promovat își pierde valabilitatea. Eliberarea atestatului nu este condiționată de promovarea examenului de bacalaureat.

Examenul național de bacalaureat la disciplina informatică

Disciplina informatică – specializarea matematică-infomatică și matematică-infomatică, intensiv informatică

Examenul național de bacalaureat 2024 se va desfășura în conformitate cu prevederile Ordinului privind organizarea și desfășurarea examenului național de bacalaureat – 2024 - O.M.E. nr. 6156/2023 și ale Metodologiei de organizare și desfășurare a examenului național de bacalaureat – 2011, aprobată prin OMECTS nr. 4799/2010, cu modificările ulterioare.

Potrivit articolului 3 din O.M.E. nr. 6156/2023, programele pentru disciplinele examenului național de bacalaureat, valabile în sesiunile anului 2024, sunt cele aprobate prin O.M.E.C.T.S. nr. 4800/2010 privind aprobarea listei disciplinelor și a programelor pentru examenul de bacalaureat.

În cadrul examenului național de bacalaureat, informatica are statutul de disciplină opțională, fiind susținută la proba E. d) în funcție de filieră, profil și specializare de absolvenții de la filiera teoretică, profil real, specializările matematică-informatică (și intensiv informatică) și științe ale naturii, respectiv de absolvenții de la filiera vocațională, profil militar, specializare matematică-informatică.

Competențele de evaluat și conținuturile specifice sunt precizate în programa de examen a disciplinei, pentru specializările matematică-informatică și matematică informatică intensiv informatică, respectiv pentru specializarea științe ale naturii.

Competențe de evaluat pentru specializarea matematică informatică (și intensiv informatică):

- construirea algoritmilor corespunzători unor prelucrări elementare și reprezentarea lor în pseudocod și prin programe scrise în limbaj de programare (Pascal, C sau C++, la alegere);
- analiza rezolvării unei probleme prin urmărirea evoluției valorilor variabilelor prelucrate de algoritmul corespunzător;
- abstractizarea rezolvării prin construirea unor algoritmi echivalenți;
- identificarea și utilizarea tipurilor de date predefinite specifice unui limbaj de programare;
- definirea și utilizarea unor tipuri de date proprii;
- identificarea și utilizarea operatorilor predefiniți elementari;

- identificarea și utilizarea subprogramelor predefinite elementare;
- identificarea și utilizarea regulilor sintactice specifice limbajului de programare studiat;
- definirea și apelul unor subprograme proprii cu înțelegerea mecanismelor de transfer prin intermediul parametrilor;
- identificarea proprietăților unor structuri de date necesare în rezolvarea problemelor cu ajutorul calculatorului și utilizarea unor modele de memorare a acestora;
- organizarea datelor ce intervin în rezolvarea unei probleme utilizând structuri de date adecvate;
- organizarea etapelor de prelucrare ce formează un algoritm utilizând structuri de control și module de program;
- folosirea unor metode sistematice de rezolvare pentru probleme de generare;
- analiza unor algoritmi echivalenți de rezolvare a unei probleme în vederea alegerii algoritmului optim.

Competențe de evaluat pentru specializarea științe ale naturii:

- construirea algoritmilor corespunzători unor prelucrări elementare și reprezentarea lor în pseudocod și prin programe scrise în limbaj de programare (Pascal, C sau C++, la alegere);
- analiza rezolvării unei probleme prin urmărirea evoluției valorilor variabilelor prelucrate de algoritmul corespunzător;
- abstractizarea rezolvării prin construirea unor algoritmi echivalenți;
- identificarea și utilizarea tipurilor de date predefinite specifice unui limbaj de programare;
- definirea și utilizarea unor tipuri de date proprii;
- identificarea și utilizarea operatorilor predefiniți elementari;
- identificarea și utilizarea subprogramelor predefinite elementare;
- identificarea și utilizarea regulilor sintactice specifice limbajului de programare studiat;
- identificarea proprietăților unor structuri de date necesare în rezolvarea problemelor cu ajutorul calculatorului și utilizarea unor modele de memorare a acestora;
- organizarea datelor ce intervin în rezolvarea unei probleme utilizând structuri de date adecvate;
- organizarea etapelor de prelucrare ce formează un algoritm utilizând structuri de control;
- analiza unor algoritmi echivalenți de rezolvare a unei probleme în vederea alegerii algoritmului optim.

Notă: Variantele de subiecte pentru examenul național de bacalaureat evaluează

competențele pe baza conținuturilor prevăzute în programa de examen, iar baremele de evaluare și de notare prevăd acordarea punctajelor pentru orice modalitate corectă de rezolvare a cerințelor.

I.2.3. Manuale pentru disciplina informatică

Manualele aprobate pentru disciplina informatică, la clasa a XII-a, cuprind prezentarea didactică a conținuturilor corespunzătoare programei, exemple de activități de învățare și exemple de itemi care pot fi utilizați pentru evaluare. Manualele care pot fi utilizate sunt cuprinse în Catalogul manualelor școlare valabile în învățământul preuniversitar în anul școlar 2023- 2024.

Titlul manualului	Autorii	Editura	Ordin nr.
Informatică	Marcel Andrei Homorodean, Simona Petrescu	Niculescu ABC	1561/2007
Informatică	R. Vișineanu, Constantin Scheau	Petrion	1561/2007
Informatică – varianta Visual Fox Pro	Mariana Paniru, Ionuț Pațiru, Irina Pațiru	L & S Info-Mat	1561/2007
Informatică teoretic, profil real, matematică-informatică	Mariana Pațiru	Editura All	1262/2007
Informatică teoretic, profil real, matematică-informatică	Alin Burța	Editura All	1262/2007
Informatică teoretic, profil real, matematică-informatică	Mioara Gheorghe, Monica Tătărâm, Corina Achinca	Corint	1342/2007
Informatică teoretic, profil real, matematică-informatică	Mariana Miloșescu	EDP	1342/2007
Informatică teoretic, profil real, matematică-informatică	Mariana Miloșescu, Oana Miloșescu	EDP	1342/2007
Informatică teoretic, profil real, matematică-informatică	Cristiana Cărnaț, Sanda Junea, Maria Scripcă	Gimnasium	1561/2007
Informatică teoretic, profil real, matematică-informatică	Marcu Daniela, Popa Carmen, Zotic Cristina	GIL	1561/2007
Informatică teoretic, profil real, matematică-informatică	Marcu Ovidiu, Zotic Cristina, Vlad Giorgie Daniel	GIL	1561/2007

Informatică teoretic, profil real, matematică-informatică	Carmen Popescu	L&S Info-Mat	1561/2007
Informatică intensiv teoretic, profil real, matematică-informatică	Mariana Miloşescu	EDP	1342/2007
Informatică intensiv teoretic, profil real, matematică-informatică	Marcu Daniela, Marcu Ovidiu, Popa Carmen, Zotic Cristina, Vlad Giorgie Daniel	GIL	1561/2007
Informatică intensiv teoretic, profil real, matematică-informatică	Vlad Tudor Huţanu, Carmen Popescu	L&S Info-Mat	1561/2007

Pe pagina web dedicată (manuale.edu.ro) sunt disponibile, în format digital, manualele:

- Informatică, Manual pentru clasa a 12-a, autor Mariana Panţiru, Editura All;
- Informatică, Manual pentru clasa a 12-a, autor Alin Burţa, Editura All.

I.3. Structura anului şcolar 2024-2025

Având în vedere prevederile O.M.E. nr. 3694/2024, referitoare la structura anului şcolar 2024-2025, pentru învăţământul liceal, clasa a XII-a, anul şcolar 2024-2025 are o durată de 34 de săptămâni de cursuri, care încep la data de 9 septembrie 2024 şi se încheie la data de 6 iunie 2025.

Anul şcolar 2024-2025, se structurează pe cinci intervale de cursuri şi intervale de vacanţe, astfel:

- **Intervalul I de cursuri:** de luni, 9 septembrie 2024, până vineri, 25 octombrie 2024;
Vacanţă: de sâmbătă, 26 octombrie 2024, până duminică, 3 noiembrie 2024;
- **Intervalul al II-lea de cursuri:** de luni, 4 noiembrie 2024, până vineri, 20 decembrie 2024;
Vacanţă: de sâmbătă, 21 decembrie 2024, până marţi, 7 ianuarie 2025;
- **Intervalul al III-lea de cursuri:** de miercuri, 8 ianuarie 2025, până vineri, 7 februarie 2025, respectiv vineri, 14 februarie 2025, sau vineri, 21 februarie 2025, după caz, la decizia inspectoratelor şcolare judeţene/al municipiului Bucureşti;
Vacanţă: o săptămână, la decizia inspectoratelor şcolare judeţene/al municipiului Bucureşti, în perioada 10 februarie - 2 martie 2025;

-
- **Intervalul al IV-lea de cursuri:** de luni, 17 februarie 2025, respectiv luni, 24 februarie 2025, sau luni, 3 martie 2025, la decizia inspectoratelor școlare județene/al municipiului București, după caz, până joi, 17 aprilie 2025;

Vacanță: de vineri, 18 aprilie 2025, până duminică, 27 aprilie 2025;

- **Intervalul al V-lea de cursuri:** de luni, 28 aprilie 2025, până vineri, 20 iunie 2025;

Vacanță: de sâmbătă, 21 iunie 2025, până duminică, 7 septembrie 2025.

În conformitate cu prevederile art. 4 alin. (1) din O.M.E. nr. 3694/2024 privind structura anului școlar 2024-2025, în perioada 9 septembrie 2024 - 30 mai 2025, se desfășoară Programul național „Școala altfel” și Programul „Săptămâna verde”, în intervale de câte 5 zile consecutive lucrătoare, a căror planificare se află la decizia unității de învățământ. Derularea celor două programe se planifică în intervale de cursuri diferite.

I.3.1. Exemple orientative de planificări calendaristice anuale

1.3.1.1. Exemplu orientativ de planificare calendaristică anuală pentru specializările matematică-informatică și matematică-informatică, intensiv informatică - MODULUL 1 – BAZE DE DATE

Unitatea de învățământ _____

Disciplina: informatică

Profesor _____

Clasa a XII-a

Nr. ore pe săpt.: 1 oră (curs) - MODULUL 1 – BAZE DE DATE

Avizat,

Director _____

PLANIFICARE CALENDARISTICĂ

Anul școlar 2024-2025

Programa aprobată cu O.M.E.C.I. nr. 5099/2009

Filiera teoretică, profil real, specializarea: matematică-informatică /matematică-informatică, intensiv informatică

Pentru varianta:

Interval de cursuri I (7 săptămâni, 09.09 – 25.10.2024)

Interval de cursuri II (7 săptămâni, 04.11 – 20.12.2024)

Interval de cursuri III (6 săptămâni, 08.01 –14.02.2025)

Interval de cursuri IV (1+7 săptămâni, 24.02 – 17.04.2025)

Interval de cursuri IIV (1+5 săptămâni, 28.04 – 06.06.2025)

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Recapitulare. Evaluare inițială/predictivă			2	S1-S2	
Modelul conceptual al unei probleme de gestiune	1.1. 1.2.	Modelul conceptual al problemei (entități, proprietăți, comportamente) Relații între entități <ul style="list-style-type: none"> • 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 	5	S3-S7	Interval de cursuri I (7 săptămâni, 09.09 – 25.10.2024)
Vacanță 26 octombrie 2024 – 3 noiembrie 2024					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Modelul conceptual al unei probleme de gestiune	1.1. 1.2.	Modelul conceptual al problemei (entități, proprietăți, comportamente) Relații între entități <ul style="list-style-type: none"> • 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 	5	S1-S5	Interval de cursuri II (7 săptămâni, 04.11 – 20.12.2024)

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Baze de date	1.4. 2.1.	Operații elementare de prelucrare a datelor Tehnici de prelucrare a datelor <ul style="list-style-type: none"> Modele de baze de date (modelul relațional, modelul rețea, modelul ierarhic) Operații specifice prelucrării bazelor de date (interogări, rapoarte) 	2	S6-S7	
Vacanță 21 decembrie 2024 – 07 ianuarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Baze de date	1.4. 2.1.	Operații elementare de prelucrare a datelor Tehnici de prelucrare a datelor <ul style="list-style-type: none"> Relaționare, cheie primară, chei externe Reguli de integritate Programe de validare, de acțiune Operații specifice prelucrării bazelor de date (interogări, rapoarte) 	2	S1-S2	Interval de cursuri III (6 săptămâni, 08.01 – 14.02.2025)
Tabele	1.3. 1.4.	Modele de organizare a datelor Structuri de date Operații elementare de prelucrare a datelor <ul style="list-style-type: none"> Crearea structurii tabelor (tipuri de date, structură, câmpuri/coloane) Conținutul unei tabele (linii/înregistrări) 	4	S3-S6	

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
		<ul style="list-style-type: none"> Operații specifice prelucrării tabelor (adăugare, modificare, ștergere, sortare, căutare, vizualizare, calcule statistice) 			
Vacanță 15 februarie 2025 – 23 februarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Programul „Școala altfel” 24 februarie – 28 februarie				S1	
Introducere în SQL; structura comenzilor SQL	2.1. 2.2. 2.3. 3.1.	Instrucțiuni specifice limbajului de programare Structura unei aplicații Modularizarea aplicației Criterii de eficiență a aplicațiilor Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice) <ul style="list-style-type: none"> Selecție, proiecție Interogări simple Inserarea, modificarea, ștergerea datelor în tabele Crearea și modificarea structurii tabelor 	7	S2-S8	Interval de cursuri IV (1+7 săptămâni, 24.02 – 17.04.2025)
Vacanță 18 aprilie 2025 – 27 aprilie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Dezvoltarea profesională în domeniul IT	3.2.	Etape în dezvoltarea aplicațiilor Reguli elementare pentru crearea și susținerea unei prezentări publice <ul style="list-style-type: none"> ● Identificarea aptitudinilor pentru anumite tipuri de activități ● Crearea unui CV și reguli de susținere a unui interviu ● Reguli în susținerea unei prelegeri ● Principii de lucru în echipă 	2	S1 – S2	Interval de cursuri V (1+5 săptămâni, 28.04 – 20.06.2025)
	3.3.				
Recapitulare. Evaluare finală/sumativă			3	S3 – S5	
Programul „Săptămâna verde” (02 iunie – 05 iunie 2025)				S6	
Examenul național de bacalaureat					

COMPETENȚE SPECIFICE:

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

- 1.1. Analizarea unei probleme în scopul identificării și clasificării datelor necesare
- 1.2. Identificarea relațiilor dintre date
- 1.3. Identificarea modalităților adecvate de structurare a datelor care intervin într-o problemă
- 1.4. Utilizarea funcțiilor specifice de prelucrare a datelor structurate

2. *Elaborarea algoritmilor de rezolvare a problemelor*

- 2.1. Identificarea tehnicilor de programare adecvate rezolvării unei probleme și aplicarea creativă a acestora
- 2.2. Elaborarea strategiei de rezolvare a unei probleme
- 2.3. Analizarea comparativă a eficienței diferitelor tehnici de rezolvare a problemei respective și alegerea celei mai eficiente variante

3. Implementarea algoritmilor într-un limbaj de programare

3.1. Utilizarea instrumentelor de dezvoltare a unei aplicații

3.2. Elaborarea și realizarea unei aplicații, folosind un mediu de programare specific

3.3. Prezentarea unei aplicații Reguli elementare pentru crearea și susținerea unei prezentări publice

1.3.1.2. Exemplu orientativ de planificare calendaristică anuală pentru specializările matematică-informatică și matematică-informatică, intensiv informatică - MODULUL 2 – SISTEME DE GESTIUNE A BAZELOR DE DATE

Unitatea de învățământ _____

Disciplina: informatică

Profesor _____

Clasa a XII-a

Nr. ore pe săptăm.: 3 ore (activități practice, cu clasa pe grupe) - **MODULUL 2 – SISTEME DE GESTIUNE A BAZELOR DE DATE**

Avizat,

Director _____

PLANIFICARE CALENDARISTICĂ

Anul școlar 2024-2025

Programa aprobată cu O.M.E.C.I. nr. 5099/2009

Filiera teoretică, profil real, specializarea: matematică-informatică /matematică-informatică, intensiv informatică

Pentru varianta:

Intervalul de cursuri I (7 săptămâni, 09.09 – 25.10.2024)

Intervalul de cursuri II (7 săptămâni, 04.11 – 20.12.2024)

Intervalul de cursuri III (6 săptămâni, 08.01 –14.02.2025)

Intervalul de cursuri IV (1+7 săptămâni, 24.02 – 17.04.2025)

Intervalul de cursuri V (1+5 săptămâni, 28.04 – 06.06.2025)

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Recapitulare. Evaluare inițială/predictivă			6	S1-S2	
Modelul conceptual al unei probleme	1.1. 1.2.	Modelul conceptual al problemei (entități, proprietăți, comportamente) Relații între entități <ul style="list-style-type: none"> • Convenții pentru realizarea diagramelor ERD • Tipuri și subtipuri • Transferabilitate 	15	S3-S7	Intervalul de cursuri I (7 săptămâni, 09.09 – 25.10.2024)
Vacanță 26 octombrie 2024 – 3 noiembrie 2024					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Modelul conceptual al unei probleme	1.1. 1.2.	Modelul conceptual al problemei (entități, proprietăți, comportamente) Relații între entități <ul style="list-style-type: none"> • Relații ierarhice, relații recursive • Arce 	15	S1-S5	Intervalul de cursuri II (7 săptămâni, 04.11 – 20.12.2024)

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Baze de date Mediul de dezvoltare a aplicațiilor	1.4. 2.1.	Operații elementare de prelucrare a datelor Tehnici de prelucrare a datelor <ul style="list-style-type: none"> • Servere de baze de date • Instalarea serverului de baze de date Oracle Database XIE și Oracle Application Express, SQLServer 	6	S6-S7	
Vacanță 21 decembrie 2024 – 07 ianuarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Introducere în SQL	2.1 2.2 2.3 3.1	Instrucțiuni specifice limbajului de programare Structura unei aplicații Modularizarea aplicației Criterii de eficiență a aplicațiilor Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice) <ul style="list-style-type: none"> • Expresii, funcții • Gruparea datelor • Sortarea datelor 	6	S1-S2	Intervalul de cursuri III (6 săptămâni, 08.01 – 14.02.2025)

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Tabele	1.3. 1.4.	Modele de organizare a datelor Structuri de date Operații elementare de prelucrare a datelor <ul style="list-style-type: none"> ● Relaționarea tabelor ● Interogări multiple (join) ● Constrângeri 	12	S3-S6	
Vacanță 15 februarie 2025 – 23 februarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Programul „Școala altfel” 24 februarie – 28 februarie				S1	
Comenzi SQL	2.1. 2.2. 2.3. 3.1.	Instrucțiuni specifice limbajului de programare Structura unei aplicații Modularizarea aplicației Criterii de eficiență a aplicațiilor Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice) <ul style="list-style-type: none"> ● Selecție, proiecție ● Interogări simple ● Inserarea, modificarea, ștergerea datelor în tabele ● Crearea și modificarea structurii tabelor 	21	S2-S8	Intervalul de cursuri IV (1+7 săptămâni, 24.02 – 17.04.2025)
Vacanță 18 aprilie 2025 – 27 aprilie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Acces la o bază de date. Formulare și rapoarte	3.2.	Etape în dezvoltarea aplicațiilor Reguli elementare pentru crearea și susținerea unei prezentări publice <ul style="list-style-type: none"> ● Gestionarea drepturilor de acces la baza de date ● Gestionarea tranzacțiilor ● Lucrul în echipă pentru elaborarea unui proiect după un plan dat ● Realizarea modelului conceptual, harta relațiilor ● Construirea și implementarea bazei de date 	6	S1 – S2	Intervalul de cursuri V (1+5 săptămâni, 28.04 – 20.06.2025)
	3.3.				
Recapitulare. Evaluare finală/sumativă			9	S3 – S5	
Programul „Săptămâna verde” (02 iunie – 05 iunie 2025)				S6	
Examenul național de bacalaureat					

COMPETENȚE SPECIFICE:

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

- 1.1. Analizarea unei probleme în scopul identificării și clasificării datelor necesare
- 1.2. Identificarea relațiilor dintre date
- 1.3. Identificarea modalităților adecvate de structurare a datelor care intervin într-o problemă
- 1.4. Utilizarea funcțiilor specifice de prelucrare a datelor structurate

2. Elaborarea algoritmilor de rezolvare a problemelor

- 2.1. Identificarea tehnicilor de programare adecvate rezolvării unei probleme și aplicarea creativă a acestora
- 2.2. Elaborarea strategiei de rezolvare a unei probleme

2.3. Analizarea comparativă a eficienței diferitelor tehnici de rezolvare a problemei respective și alegerea celei mai eficiente variante

3. Implementarea algoritmilor într-un limbaj de programare

3.1. Utilizarea instrumentelor de dezvoltare a unei aplicații

3.2. Elaborarea și realizarea unei aplicații, folosind un mediu de programare specific

3.3. Prezentarea unei aplicații Reguli elementare pentru crearea și susținerea unei prezentări publice

1.3.1.3. Exemplu orientativ de planificare calendaristică anuală pentru specializările matematică-informatică și matematică-informatică, intensiv informatică - MODULUL 3. PROGRAMARE VIZUALĂ

Unitatea de învățământ _____

Disciplina: informatică

Profesor _____

Clasa a XII-a

Nr. ore pe săptăm.: 3 ore (1 oră curs+2 ore aplicații practice cu toată clasa) - **MODULUL 3. PROGRAMARE VIZUALĂ**

Avizat,

Director _____

PLANIFICARE CALENDARISTICĂ

Anul școlar 2024-2025

Programa aprobată cu O.M.E.C.I. nr. 5099/2009

Filiera teoretică, profil real, specializarea: matematică-informatică /matematică-informatică, intensiv informatică

Pentru varianta:

Intervalul de cursuri I (7 săptămâni, 09.09 – 25.10.2024)

Intervalul de cursuri II (7 săptămâni, 04.11 – 20.12.2024)

Intervalul de cursuri III (6 săptămâni, 08.01 –14.02.2025)

Intervalul de cursuri IV (1+7 săptămâni, 24.02 – 17.04.2025)

Intervalul de cursuri V (1+5 săptămâni, 28.04 – 06.06.2025)

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Recapitulare. Evaluare inițială/predictivă			6	S1-S2	
Introducere în programarea vizuală	1.1.	Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date	3	S3	Intervalul de cursuri I (7 săptămâni, 09.09 – 25.10.2024)
	2.1.	Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice). Etape în dezvoltarea aplicațiilor			
2.3.	Tehnici de prelucrare a datelor. Instrucțiuni specifice limbajului de programare.				
3.1.	Criterii de eficiență a aplicațiilor				
3.2.	Reguli elementare pentru crearea și susținerea unei prezentări publice				
3.3.	<ul style="list-style-type: none"> ● Concepte de bază ale programării vizuale ● Prezentarea unui mediu de programare vizual (Microsoft Visual C#, [...]). Operații și unelte specifice ale acestuia ● Bare de unelte <p>Elemente de programare orientată pe obiecte în context vizual</p>				
Construirea interfeței utilizator – operare cu formulare și butoane	1.1.	Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date. Relații între entități	12	S4-S7	
	1.2.	Modele de organizare a datelor. Structuri de date			
	1.3.	Operații elementare de prelucrare a datelor			
	1.4.	Structura unei aplicații. Modularizarea aplicației			
	2.2.	<ul style="list-style-type: none"> ● Construirea interfeței utilizator ● Ferestre: tipuri, utilizare, organizare, proprietăți, evenimente <p>Dezvoltarea și prezentarea unei aplicații în mediu vizual</p>			
Vacanță 26 octombrie 2024 – 3 noiembrie 2024					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Construirea interfeței utilizator – operare cu imagini	1.1. 1.2. 1.3. 1.4. 2.2.	<p>Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date. Relații între entități</p> <p>Modele de organizare a datelor. Structuri de date</p> <p>Operații elementare de prelucrare a datelor</p> <p>Structura unei aplicații. Modularizarea aplicației</p> <ul style="list-style-type: none"> ● Controale: tipuri, utilizare, organizare, proprietăți, evenimente ● Dezvoltarea și prezentarea unei aplicații în mediu vizual 	9	S1-S3	<p>Intervalul de cursuri II</p> <p><i>(7 săptămâni, 04.11 – 20.12.2024)</i></p>
Construirea interfeței utilizator – operare cu text	1.1. 1.2. 1.3. 1.4. 2.2.	<p>Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date. Relații între entități</p> <p>Modele de organizare a datelor. Structuri de date</p> <p>Operații elementare de prelucrare a datelor</p> <p>Structura unei aplicații. Modularizarea aplicației</p> <ul style="list-style-type: none"> ● Controale: tipuri, utilizare, organizare, proprietăți, evenimente ● Dezvoltarea și prezentarea unei aplicații în mediu vizual 	12	S4-S7	
Vacanță 21 decembrie 2024 – 07 ianuarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Construirea interfeței utilizator – operarea cu liste și meniuri	1.1. 1.2. 1.3. 1.4. 2.2.	Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date. Relații între entități Modele de organizare a datelor. Structuri de date Operații elementare de prelucrare a datelor Structura unei aplicații. Modularizarea aplicației <ul style="list-style-type: none"> ● Controale: tipuri, utilizare, organizare, proprietăți, evenimente ● Meniuri ● Dezvoltarea și prezentarea unei aplicații în mediu vizual 	6	S1-S2	Intervalul de cursuri III <i>(6 săptămâni,</i> <i>08.01 –</i> <i>14.02.2025)</i>
Construirea interfeței utilizator – operarea cu dialoguri și ferestre predefinite	1.1. 1.2. 1.3. 1.4. 2.2.	Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date. Relații între entități Modele de organizare a datelor. Structuri de date Operații elementare de prelucrare a datelor Structura unei aplicații. Modularizarea aplicației <ul style="list-style-type: none"> ● Controale: tipuri, utilizare, organizare, proprietăți, evenimente ● Obiecte grafice ● Dezvoltarea și prezentarea unei aplicații în mediu vizual 	3	S3	
Construirea interfeței utilizator în run-time	1.1. 1.2. 1.3. 1.4. 2.2.	Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date. Relații între entități Modele de organizare a datelor. Structuri de date Operații elementare de prelucrare a datelor Structura unei aplicații. Modularizarea aplicației <ul style="list-style-type: none"> ● Controale: proprietăți, evenimente ● Dezvoltarea și prezentarea unei aplicații în mediu vizual 	9	S4-S6	
Vacanță 15 februarie 2025 – 23 februarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Programul „Școala altfel” 24 februarie – 28 februarie				S1	
Construirea interfeței utilizator – controale pentru cronometrare și urmărire	1.1. 1.2. 1.3. 1.4. 2.2.	Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date. Relații între entități Modele de organizare a datelor. Structuri de date Operații elementare de prelucrare a datelor Structura unei aplicații. Modularizarea aplicației <ul style="list-style-type: none"> ● Controale: tipuri, utilizare, organizare, proprietăți, evenimente ● Dezvoltarea și prezentarea unei aplicații în mediu vizual 	9	S2-S4	Intervalul de cursuri IV (1+7 săptămâni, 24.02 – 17.04.2025)
Construirea interfeței utilizator – controale pentru desenare	1.1. 1.2. 1.3. 1.4. 2.2.	Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date. Relații între entități Modele de organizare a datelor. Structuri de date Operații elementare de prelucrare a datelor Structura unei aplicații. Modularizarea aplicației <ul style="list-style-type: none"> ● Controale: tipuri, utilizare, organizare, proprietăți, evenimente ● Obiecte grafice ● Dezvoltarea și prezentarea unei aplicații în mediu vizual 	12	S5-S8	
Vacanță 18 aprilie 2025 – 27 aprilie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Baze de date în context vizual	1.1.	Modelul conceptual al problemei (entități, proprietăți, comportamente). Tipuri de date. Relații între entități	6	S1-S2	Intervalul de cursuri V (1+5 săptămâni, 28.04 – 20.06.2025)
	1.2.	Modele de organizare a datelor. Structuri de date			
	1.3.	Operații elementare de prelucrare a datelor			
	1.4.	Structura unei aplicații. Modularizarea aplicației			
	2.2.	<ul style="list-style-type: none"> ● Controale: tipuri, utilizare, organizare, proprietăți, evenimente ● 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 			
Recapitulare. Evaluare finală/sumativă			9	S3 – S5	
Programul „Săptămâna verde” (02 iunie – 05 iunie 2025)				S6	
Examenul național de bacalaureat					

COMPETENȚE SPECIFICE:

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

- 1.1. Analizarea unei probleme în scopul identificării și clasificării datelor necesare
- 1.2. Identificarea relațiilor dintre date
- 1.3. Identificarea modalităților adecvate de structurare a datelor care intervin într-o problemă
- 1.4. Utilizarea funcțiilor specifice de prelucrare a datelor structurate

2. Elaborarea algoritmilor de rezolvare a problemelor

- 2.1. Identificarea tehnicilor de programare adecvate rezolvării unei probleme și aplicarea creativă a acestora
- 2.2. Elaborarea strategiei de rezolvare a unei probleme
- 2.3. Analizarea comparativă a eficienței diferitelor tehnici de rezolvare a problemei respective și alegerea celei mai eficiente variante

3. Implementarea algoritmilor într-un limbaj de programare

- 3.1. Utilizarea instrumentelor de dezvoltare a unei aplicații
- 3.2. Elaborarea și realizarea unei aplicații, folosind un mediu de programare specific
- 3.3. Prezentarea unei aplicații Reguli elementare pentru crearea și susținerea unei prezentări publice

1.3.1.4. Exemplu orientativ de planificare calendaristică anuală pentru specializările matematică-informatică și matematică-informatică, intensiv informatică - MODULUL 4. PROGRAMARE WEB

Unitatea de învățământ _____

Disciplina: informatică

Profesor _____

Clasa a XII-a

Nr. ore pe săpt.: 3 ore (1 oră curs+2 ore aplicații practice cu toată clasa) - **MODULUL 4. PROGRAMARE WEB**

Avizat,

Director _____

PLANIFICARE CALENDARISTICĂ

Anul școlar 2024-2025

Programa aprobată cu O.M.E.C.I. nr. 5099/2009

Filiera teoretică, profil real, specializarea: matematică-informatică /matematică-informatică, intensiv informatică

Pentru varianta:

Intervalul de cursuri I (7 săptămâni, 09.09 – 25.10.2024)

Intervalul de cursuri II (7 săptămâni, 04.11 – 20.12.2024)

Intervalul de cursuri III (6 săptămâni, 08.01 –14.02.2025)

Intervalul de cursuri IV (1+7 săptămâni, 24.02 – 17.04.2025)

Intervalul de cursuri V (1+5 săptămâni, 28.04 – 06.06.2025)

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Recapitulare. Evaluare inițială/predictivă			6	S1-S2	
Principii generale ale proiectării interfețelor web -I	1.1.	Modelul conceptual al problemei (entități, proprietăți, comportamente)	15	S3-7	Intervalul de cursuri I (7 săptămâni, 09.09 – 25.10.2024)
		Tipuri de date			
		Relații între entități			
	1.2.	Modele de organizare a datelor			
	1.3.	Structuri de date			
1.4.	Operații elementare de prelucrare a datelor				
		<ul style="list-style-type: none"> • 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 			
Vacanță 26 octombrie – 3 noiembrie 2024					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Principii generale ale proiectării interfețelor web - II	1.1.	Modelul conceptual al problemei (entități, proprietăți, comportamente)	6	S1 – S2	Intervalul de cursuri II (7 săptămâni, 04.11 – 20.12.2024)
		Tipuri de date			
	1.2.	Relații între entități			
		Modele de organizare a datelor			
	1.3.	Structuri de date			
	1.4.	Operații elementare de prelucrare a datelor			
	3.1.	Realizarea interfețelor Web utilizând limbajul de marcare HTML (elemente avansate):			
3.2.	<ul style="list-style-type: none"> • Foi de stiluri (CSS) 				

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Modelul client-server. Protocoale de comunicație	1.3. 3.1.	Modele de organizare a datelor <ul style="list-style-type: none"> ● Modelul client-server. ● Protocoale de comunicație 	3	S3	
Mediul de lucru	3.1.	Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice) <ul style="list-style-type: none"> ● Mediul de lucru (server web - Apache, IIS etc., instrumente de dezvoltare a aplicațiilor - PHPdev etc.) 	3	S4	
Limbaje de scripting server-side	2.1. 2.2. 2.3. 3.1.	Tehnici de prelucrare a datelor Instrucțiuni specifice limbajului de programare <ul style="list-style-type: none"> ● Prezentarea limbajului de scripting - PHP, ASP etc. ● Elemente de bază ale limbajului ● Instrucțiuni 	9	S5-S7	
Vacanță 21 decembrie 2024 – 07 ianuarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Limbaje de scripting server-side	2.1. 2.2. 2.3. 3.1.	Tehnici de prelucrare a datelor Instrucțiuni specifice limbajului de programare Prezentarea limbajului de scripting - PHP, ASP etc. <ul style="list-style-type: none"> ● Funcții ● Structuri de date 	18	S1-S6	Intervalul de cursuri III (6 săptămâni, 08.01 – 14.02.2025)
Vacanță 15 februarie 2025 – 23 februarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Programul „Școala altfel” 24 februarie – 28 februarie				S1	Intervalul de cursuri IV (1+7 săptămâni, 24.02 – 17.04.2025)
Interacțiunea cu baze de date web	1.1. 1.2. 1.3. 1.4. 3.1.	Modele de organizare a datelor Structuri de date Interacțiunea cu baze de date Web <ul style="list-style-type: none"> • 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 	18	S2-S8	
Vacanță 18 aprilie 2025 – 27 aprilie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Proiectarea și realizarea unei aplicații web	1.1. 1.2. 1.3. 2.1. 2.2. 2.3. 3.1. 3.2. 3.3.	Proiectarea și realizarea unei aplicații Web Etape în dezvoltarea aplicațiilor Structura unei aplicații Modularizarea aplicației Criterii de eficiență a aplicațiilor Reguli elementare pentru crearea și susținerea unei prezentări publice	6	S1 – S2	Intervalul de cursuri V (1+5 săptămâni, 28.04 – 20.06.2025)
Recapitulare. Evaluare finală/sumativă			9	S3 – S5	
Programul „Săptămâna verde” (02 iunie – 05 iunie 2025)				S6	
Examenul național de bacalaureat					

COMPETENȚE SPECIFICE:

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

- 1.1. Analizarea unei probleme în scopul identificării și clasificării datelor necesare
- 1.2. Identificarea relațiilor dintre date
- 1.3. Identificarea modalităților adecvate de structurare a datelor care intervin într-o problemă
- 1.4. Utilizarea funcțiilor specifice de prelucrare a datelor structurate

2. Elaborarea algoritmilor de rezolvare a problemelor

- 2.1. Identificarea tehnicilor de programare adecvate rezolvării unei probleme și aplicarea creativă a acestora
- 2.2. Elaborarea strategiei de rezolvare a unei probleme
- 2.3. Analizarea comparativă a eficienței diferitelor tehnici de rezolvare a problemei respective și alegerea celei mai eficiente variante

3. Implementarea algoritmilor într-un limbaj de programare

- 3.1. Utilizarea instrumentelor de dezvoltare a unei aplicații
- 3.2. Elaborarea și realizarea unei aplicații, folosind un mediu de programare specific
- 3.3. Prezentarea unei aplicații Reguli elementare pentru crearea și susținerea unei prezentări publice

1.3.1.5. Exemplu orientativ de planificare calendaristică anuală pentru specializările matematică-informatică și matematică-informatică, intensiv informatică - MODULUL 5 – PROGRAMAREA PROCEDURALĂ A BAZELOR DE DATE (PL/SQL, TransactSQL, MySQL)

Unitatea de învățământ _____

Disciplina: Informatică

Profesor _____

Clasa a XII-a

Nr. ore pe săpt.: 3 ore (activități practice, cu clasa pe grupe) - **MODULUL 5 – PROGRAMAREA PROCEDURALĂ A BAZELOR DE DATE (PL/SQL, TransactSQL, MySQL)**

Avizat,

Director _____

PLANIFICARE CALENDARISTICĂ

Anul școlar 2024-2025

Programa aprobată cu O.M.E.C.I. nr. 5099/2009

Filiera teoretică, profil real, specializarea: matematică-informatică /matematică-informatică, intensiv informatică

Pentru varianta:

Intervalul de cursuri I (7 săptămâni, 09.09 – 25.10.2024)

Intervalul de cursuri II (7 săptămâni, 04.11 – 20.12.2024)

Intervalul de cursuri III (6 săptămâni, 08.01 – 14.02.2025)

Intervalul de cursuri IV (1+7 săptămâni, 24.02 – 17.04.2025)

Intervalul de cursuri V (1+5 săptămâni, 28.04 – 06.06.2025)

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Recapitulare. Evaluare inițială/predictivă			6	S1-S2	
Elemente ale limbajului	1.1.	Modelul conceptual al problemei (entități, proprietăți, comportamente)	9	S3-S5	Intervalul de cursuri I (7 săptămâni, <i>09.09 – 25.10.2024</i>)
	1.2.	Tipuri de date Relații între entități <ul style="list-style-type: none"> • Identificatori, tipuri de date, variabile. • Operatori. 			
Elemente de programare - I	2.1.	Tehnici de prelucrare a datelor Instrucțiuni specifice limbajului de programare <ul style="list-style-type: none"> • Structuri de control. 	6	S6-S7	
Vacanță 26 octombrie 2024 – 3 noiembrie 2024					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Elemente de programare - II	2.2.	Structura unei aplicații Modularizarea aplicației <ul style="list-style-type: none"> • Proceduri stocate. • Funcții. • Cursoare. • Triggers. 	21	S1-S7	Intervalul de cursuri II (7 săptămâni, <i>04.11 – 20.12.2024</i>)
Vacanță 21 decembrie 2024 – 07 ianuarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Tipuri structurate de date	1.3. 1.4.	Modele de organizare a datelor Operații elementare de prelucrare a datelor Structuri de date <ul style="list-style-type: none"> • Tipuri structurate de date 	6	S1-S2	Intervalul de cursuri III <i>(6 săptămâni, 08.01 – 14.02.2025)</i>
Tratarea excepțiilor și gestionarea tranzacțiilor	1.3. 2.2.	Modele de organizare a datelor Structuri de date Structura unei aplicații Modularizarea aplicației <ul style="list-style-type: none"> • Gestiuinea tranzacțiilor • Tratarea excepțiilor 	12	S3-S6	
Vacanță 15 februarie 2025 – 23 februarie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Programul „Școala altfel” 24 februarie – 28 februarie				S1	Intervalul de cursuri IV (1+7 <i>săptămâni, 24.02 – 17.04.2025</i>)
Realizarea unui proiect de gestiune a datelor specifice unui domeniu de interes practic	3.1. 3.2.	Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice) Etape în dezvoltarea aplicațiilor <ul style="list-style-type: none"> • Realizarea unui proiect de gestiune a datelor specifice unui domeniu de interes practic 	3	S7	
Vacanță 18 aprilie 2025 – 27 aprilie 2025					

Unitatea de învățare	Competențe specifice	Conținuturi	Număr de ore alocate	Săptămâna	Observații
Pregătirea examenului de certificare competențelor	2.3.	Mediul de dezvoltare a aplicațiilor (interfață, instrumente specifice)	6	S1 – S2	Intervalul de cursuri V (1+5 săptămâni, 28.04 – 20.06.2025)
	3.1.	Etape în dezvoltarea aplicațiilor			
	3.2.	Criterii de eficiență a aplicațiilor			
	3.3.	Reguli elementare pentru crearea și susținerea unei prezentări publice <ul style="list-style-type: none"> Realizarea unui proiect de gestiune a datelor specifice unui domeniu de interes practic 			
Recapitulare. Evaluare finală/sumativă			9	S3 – S5	
Programul „Săptămâna verde” (02 iunie – 05 iunie 2025)				S6	
Examenul național de bacalaureat					

COMPETENȚE SPECIFICE:

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

- 1.1. Analizarea unei probleme în scopul identificării și clasificării datelor necesare
- 1.2. Identificarea relațiilor dintre date
- 1.3. Identificarea modalităților adecvate de structurare a datelor care intervin într-o problemă
- 1.4. Utilizarea funcțiilor specifice de prelucrare a datelor structurate

2. *Elaborarea algoritmilor de rezolvare a problemelor*

- 2.1. Identificarea tehnicilor de programare adecvate rezolvării unei probleme și aplicarea creativă a acestora
- 2.2. Elaborarea strategiei de rezolvare a unei probleme
- 2.3. Analizarea comparativă a eficienței diferitelor tehnici de rezolvare a problemei respective și alegerea celei mai eficiente variante

3. Implementarea algoritmilor într-un limbaj de programare

3.1. Utilizarea instrumentelor de dezvoltare a unei aplicații

3.2. Elaborarea și realizarea unei aplicații, folosind un mediu de programare specific

3.3. Prezentarea unei aplicații Reguli elementare pentru crearea și susținerea unei prezentări publice

I.4. Orientări privind proiectarea didactică

În proiectarea didactică (planificarea calendaristică și proiectarea unităților de învățare) se va ține cont și de rezultatele obținute la evaluarea inițială/predictivă realizată de profesor la începutul anului școlar/unității de învățare. Evaluarea inițială/predictivă are un rol reglator, oferind repere pentru o proiectare curriculară autentică și realistă în clasa a XII-a pe baza unor analize documentate.

Programa școlară este astfel concepută, încât să încurajeze creativitatea didactică și adecvarea demersurilor didactice la particularitățile elevilor.

Conceptul central al proiectării didactice este demersul didactic personalizat, iar instrumentul acestuia este unitatea de învățare. Demersul didactic personalizat exprimă dreptul profesorului de a lua decizii asupra modalităților pe care le consideră optime în creșterea eficienței activității de predare-învățare-evaluare, respectiv, răspunderea personală pentru a asigura elevilor un parcurs școlar individualizat, în funcție de condiții și cerințe concrete. Prin proiectarea didactică sunt asociate într-un mod personalizat elementele programei – competențe specifice, conținuturi, activități de învățare – cu alocarea de resurse (de loc, de timp și materiale) considerată optimă de către profesor.

În proiectarea didactică se va ține cont de specificul clasei, de necesitatea integrării elevilor cu nevoi speciale în colectivul clasei. Astfel, se vor adapta metodele didactice, asigurând eficiența învățării pornind de la particularitățile individuale ale fiecărui elev, folosind acele metode didactice specifice învățământului integrat și strategiile învățării interactive. Se recomandă realizarea unui plan de intervenție individualizat pentru acești elevi. De asemenea, în proiectarea activității didactice se va ține cont și de elevii supradotați, capabili de performanță, prin planificarea unor activități diferențiate de studiu și /evaluare.

Informatica, prin specificul ei, este esențial legată de lucrul individual pe calculator. Pe de altă parte, prin intermediul rețelelor de calculatoare, este posibil schimbul de informații între mai mulți utilizatori de calculatoare poate mai eficient decât prin orice altă metodă clasică de comunicare. Se va avea în vedere dezvoltarea competențelor de lucru în echipă bazate pe colaborarea elevilor în contextul unor activități organizate pe grupe.

De asemenea, se recomandă aplicarea a minimum 2-3 modele de teste, structurate după varianta de subiecte de bacalaureat elaborată ca model de către C.N.P.E.E. și accesibilă prin intermediul site-ului:

<http://subiecte.edu.ro/2024/bacalaureat/modeledesubiecte/probescrise/>,

eventual avându-se în vedere și variante de subiecte utilizate la examenele din anii precedenți, pentru familiarizarea viitorilor candidați cu desfășurarea probei E. d) din cadrul examenului național de bacalaureat.

II. ORIENTAREA PROCESULUI DE PREDARE-ÎNVĂȚARE-EVALUARE LA DISCIPLINA INFORMATICĂ, AVÂND CA REFERENȚIAL PROFILUL DE FORMARE AL ABSOLVENTULUI ȘI PENTRU PREGĂTIREA ELEVILOR ÎN VEDEREA SUSȚINERII EXAMENULUI NAȚIONAL DE BACALAUREAT ȘI A EXAMENELOR DE CERTIFICARE A COMPETENȚELOR PROFESIONALE

II.1. Exemplu de activitate de învățare care să susțină formarea de competențe specifice

Modulul de programare vizuală

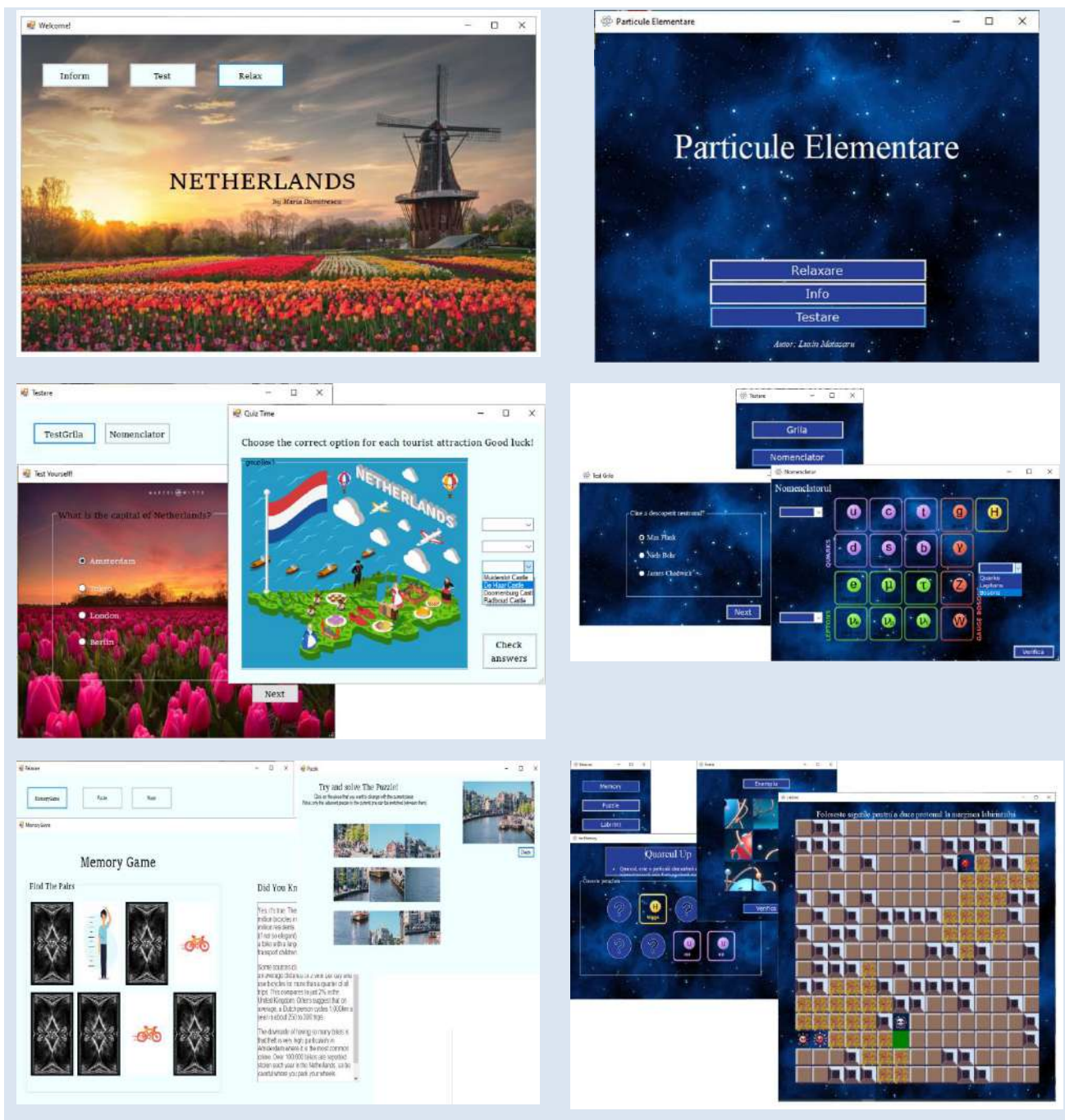
Pe parcursul clasei a XII-a, elevii dezvoltă, în cadrul lecțiilor din cadrul modulului Programare vizuală, într-o manieră multi-, inter- și transdisciplinară, o aplicație educativă ce poate fi prezentată ulterior ca proiect în cadrul examenului de atestare a competențelor profesionale. Utilizând conținuturi specifice modulului de Programare vizuală, creată în limbajul C#, cu instrumente și mecanisme specifice unui mediu de programare vizuală adecvat, aplicația educativă abordează o tematică anumită din trei puncte de vedere:

- informare – prezentarea unor informații despre tematica aleasă (filme, prezentări, imagini, adrese web etc.);
- evaluare – prin teste grilă, teste de identificare a unor elemente componente/specifice etc.;
- jocuri tematice.

Aplicația este, la rândul său, vizuală, beneficiind de elemente specifice (formulare, butoane, liste derulante etc.): din formularul principal utilizatorul poate alege una dintre cele trei perspective, apoi una dintre variantele propuse specifice.

Profesorul arată un model de astfel de aplicație la începutul studiului modulului de programare vizuală.

De exemplu:



Fiecare elev va alege o tematică proprie, având în vedere și documentarea corespunzătoare pentru tematica aleasă, prin selectarea informațiilor relevante, a materialelor multi-media în funcție de preferințele și preocupările sale.

După fiecare lecție sau grup de lecții, elevii valorifică și dezvoltă competențele formate prin integrarea în această aplicație a unui nou modul (de exemplu un nou joc, un nou test etc.).

Activitatea didactică ce urmează vizează crearea modulului Memory Game.

Titlu:

Utilizarea controalelor PictureBox pentru operarea cu imagini în programarea vizuală, în vederea dezvoltării unui joc.

Competențe specifice vizate:

- 1.1. Analizarea unei probleme în scopul identificării și clasificării datelor necesare
- 1.2. Identificarea relațiilor dintre date
- 1.3. Identificarea modalităților adecvate de structurare a datelor care intervin într-o problemă
- 1.4. Utilizarea funcțiilor specifice de prelucrare a datelor structurate
- 2.2. Elaborarea strategiei de rezolvare a unei probleme

Condiții necesare desfășurării activității:

Pe calculatoarele din laboratorul de informatică este instalată aplicația Visual Studio care permite dezvoltarea aplicațiilor de tip WindowsForm în limbajul C#.

Elevii au ales tematica aplicației educative precizate și au dezvoltat o parte a acesteia care deja cuprinde formularul principal (FormMain), formularul care conduce la aplicații de tip jocuri (FormRelaxare) și eventual alte module din cadrul acesteia. Aplicația are asociat un folder, cu numele Resurse, care cuprinde imagini și alte materiale utilizate.

Elevii sunt familiarizați cu crearea formularelor, utilizarea controalelor de tip Button, PictureBox și RichTextBox, precum și cu modalități de generare aleatoare a numerelor.

Timpul alocat:

100 de minute

Forme de organizare a clasei:

Individual

Metode didactice utilizate:

Exercițiu, expunere, modelare, învățare prin descoperire

Mijloace de învățământ utilizate:

Fișă de exerciții, prezentare, aplicații model, aplicația VisualStudio

Scenariul activității de învățare:

Activitatea profesorului:	Activitatea elevului:
Se prezintă scopul activității didactice: crearea unui joc prin care este antrenată memoria.	Elevii urmăresc prezentarea și, eventual, pun întrebări.

Pe un formular se află mai multe perechi de imagini, ca niște cărți de joc. Inițial, toate cărțile sunt cu fața în jos. La un pas oarecare, două cărți rămân cu fața în sus dacă ele au aceeași imagine și sunt accesate consecutiv, altfel se păstrează cu fața în sus doar ultima carte accesată. Dacă cele două cărți identice au fost descoperite, apare și un text explicativ corespunzător conținutului acestora.

Scopul jocului este aducerea cu fața în sus a tuturor perechilor de cărți.

Se prezintă poziția în cadrul aplicației, în care va fi integrat modulul corespunzător jocului creat, precum și un model, exemplificându-se modul de desfășurare a jocului.

Pregătirea resurselor

Se indică elevilor resursele pe care trebuie să le pregătească, conform următoarelor cerințe:

-Creați în folderul **Resurse** un subfolder, cu numele **Memory**, în care plasați resursele utilizate pentru acest joc;

-Căutați pe web o imagine corespunzătoare pentru o carte de joc întoarsă cu fața în jos, apoi trei-cinci imagini care să fie conexe cu tematica tratată în cadrul aplicației, precum și informații explicative cu privire la conținutul imaginilor alese;

-Salvați în folderul de **Resurse** asociate jocului imaginile alese (de exemplu **N**), în format **JPEG**, iar informațiile explicative asociate în fișiere în format **Rich Text Format**;

-Redenumiți fișierele cu numere nenule (**1.jpg**, **2.jpg**, ... **N.jpg**, respectiv **1.rtf**, **2.rtf**, ...**N.rtf**),

Elevii caută pe web:

-imagini adecvate tematicii alese, precum și informații explicative cu privire la conținutul acestora;

-imagine pentru cartea de joc

și apoi salvează fișierele obținute în folderul indicat, în formatul și cu numele corespunzător.

<p>iar fișierul care conține imaginea corespunzătoare unei cărți întoarse cu fața în jos, cu 0.jpg.</p>	
<p><i>Integrarea în aplicație</i></p> <p>Se indică elevilor formularul care trebuie pregătit și modul de realizare a legăturii cu restul jocului:</p> <ul style="list-style-type: none"> -Adăugați proiectului un formular nou, cu numele FormMemory - formularul care permite rularea jocului. -Stabiliți legătura cu formularul de acces la jocuri FormRelaxare, plasând pe formularul FormRelaxare un buton cu textul asociat Memory. -Tratați evenimentul Click al butonului, astfel încât să permită deschiderea formularului FormMemory. 	<p>Elevii creează formularul indicat și realizează integrarea sa în aplicația proprie.</p>
<p><i>Utilizarea controalelor specifice</i></p> <p>Se identifică, împreună cu elevii, controalele care sunt utilizate:</p> <ul style="list-style-type: none"> -Accesați formularul FormMemory, în modul Design View; -Plasați pe formular un control de tip RichTextBox și un control de tip GroupBox, în care aduceți 2·N controale de tip PictureBox; -Completați proprietatea text a controlului GroupBox cu o frază care să indice modul de operare; -Încărcați în fiecare dintre controalele de tip PictureBox imaginea corespunzătoare cărții cu fața în jos (0.jpg). <p>Se indică elevilor noi modalități de operare cu mai multe controale: selectarea acestora și apoi încărcarea simultană a imaginii.</p>	<p>Elevii participă la discuții și la identificarea controalelor necesare, apoi plasează controalele precizate pe formular.</p>

<p><i>Învățare prin descoperire</i></p> <p>De asemenea, se indică elevilor să descopere în meniu unelte pentru alinierea controalelor.</p>	<p>Elevii descoperă noi modalități de operare cu controalele, identificând în meniu uneltele specifice.</p>
<p><i>Scrierea codului</i></p> <p>Se identifică, împreună cu elevii, prin întrebări adecvate, modalitatea de operare și se deduce codul care trebuie implementat pentru funcționarea aplicației:</p> <p>-Accesați formularul <i>FormMemory</i>, în modul <i>CodeView</i>;</p> <p>-Declarați o nouă dată în clasa curentă <i>FormMemory</i>, care va memora ultima imagine <i>pictureBoxAux</i> aflată cu fața în sus.</p>	<p>Elevii participă la discuție, la identificarea etapelor algoritmului și scriu codul pentru declararea unei noi date-membru al clasei.</p> <pre>public partial class FormMemory : Form { PictureBox pictureBoxAux; public Form1() { InitializeComponent(); } }</pre>
<p>-Tratați evenimentul <i>Click</i> al primei imagini și abonați toate imaginile la handlerul de eveniment al acesteia. Acționarea imaginii trebuie să aibă ca efect întoarcerea imaginii curente cu fața în sus. Dacă imaginea aceasta este identică cu ultima imagine întoarsă, atunci se încarcă în controlul de tip <i>RichTextBox</i> textul explicativ asociat, iar cele două imagini rămân cu fața în sus, inaccesibile. În cazul în care cele două imagini sunt diferite, rămâne cu fața în sus doar ultima imagine.</p> <p>-Se indică elevilor modalitatea de abonare a tuturor imaginilor la același handler de eveniment (prin selectarea imaginilor, apoi selectarea, în fereastra de proprietăți comună, a handlerului precizat).</p> <p>-Se indică elevilor rolul și modul de utilizare a parametrului <i>sender</i> al metodei în acest context, care va fi convertit la tipul controlului corespunzător.</p>	<p>Elevii scriu codul corespunzător acțiunii de accesare a unei imagini, de exemplu:</p> <pre>private void pictureBox1_Click(object sender, EventArgs e) { (sender as PictureBox).Load("resurse//" + Convert.ToString((sender as PictureBox).Tag) + ".jpg"); if (pictureBoxAux == null) pictureBoxAux = sender as PictureBox; else if (pictureBoxAux.Tag.Equals((sender as PictureBox).Tag) == true) { pictureBoxAux.Enabled = false; (sender as PictureBox).Enabled = false; richTextBox1.LoadFile("resurse//" + Convert.ToString((sender as PictureBox).Tag) + ".rtf"); pictureBoxAux = null; } else { pictureBoxAux.Load("resurse//0.jpg"); pictureBoxAux = (sender as PictureBox); } } }</pre>

-Tratați evenimentul **Load** al formularului, pentru generarea aleatoare a perechilor de imagini.

Numărul de ordine al fiecărei imagini este memorat în proprietatea **Tag** a controlului în care este încărcată, iar alegerea acestuia se face aleator.

-Rulați aplicația și verificați funcționalitatea acesteia.

Elevii scriu codul corespunzător acțiunii de inițializare a datelor la încărcarea formularului, de exemplu:

```
private void FormMemory_Load(object sender, EventArgs e)
{
    int nrPerechi = 2, i, tag;
    int[] perechi = new int[nrPerechi+1];
    Random r = new Random();
    for(i=0; i<=nrPerechi; i++)
        perechi[i]=0;
    foreach(PictureBox p in groupBox1.Controls)
    {
        do
        {
            tag = 1+r.Next(nrPerechi);
        } while (perechi[tag] == 2);
        perechi[tag]++;
        p.Tag = tag;
    }
    pictureBoxAux = null;
}
```

Elevii rulează aplicația și verifică funcționalitatea acesteia, tratând eventualele erori.

Observații:

Activitatea didactică îmbină utilizarea competențelor specifice disciplinei informatică, cu cele specifice altor discipline, favorizând dezvoltarea și implementarea algoritmilor în diverse aplicații.

II.2. Exemple de teme pentru recapitulare/sinteză, în contextul susținerii, la finalul anului școlar, a examenelor specifice

II.2.1. Teme pentru recapitulare - divizibilitate

Tematica divizibilității apare în multe dintre subiectele de la examenul național de bacalaureat, la disciplina informatică.

Competențele de evaluat, prevăzute în programa de bacalaureat la disciplina informatică, specializările matematică-informatică, matematică-informatică, intensiv informatică, respectiv la clasele din filiera vocațională, profil militar, specializarea matematică-informatică, vor fi fixate pe baza conținuturilor specifice prevăzute în programă, corelate cu tematica divizibilității. Astfel, se recomandă recapitularea:

- noțiunilor specifice: divizor, divizor propriu, număr prim, numere prime între ele;
- algoritmilor de bază specifici pentru: parcurgerea divizorilor naturali/divizorilor proprii ai unui număr natural, determinarea celui mai mare divizor comun, descompunerea în factori primi.

Noțiuni specifice

Divizor

Un număr întreg nenul d este divizor al unui număr întreg n dacă n se poate scrie ca produsul dintre d și un alt număr întreg, d_1 .

Așadar,

- dacă $n=d \cdot d_1 \Rightarrow d, d_1$ sunt divizori ai lui n ;
- $n \% d = 0 \Leftrightarrow d$ este divizor al lui n ;
- d este divizor al lui $n \Rightarrow d_1 = n/d$ este divizor al lui n ,

unde s-a notat cu $n \% d$ restul împărțirii numărului natural n la numărul natural nenul d .

Exemplu: numărul 12 îi are ca divizori pe 1, -1, 2, -2, 3, -3, 4, -4, 6, -6, 12 și -12.

Divizor propriu

Un divizor propriu al unui număr întreg n este un divizor natural al lui n diferit de 1 și de n .

Exemplu: 12 îi are ca divizori proprii pe 2, 3, 4 și 6.

Număr prim

Un număr prim este un număr natural care are doi divizori naturali (1 și el însuși).

Exemplu: 3 și 5 sunt numere prime; 12, 0 și 1 nu sunt numere prime (nu au câte doi divizori naturali).

Numere prime între ele

Două numere sunt prime între ele dacă au un singur divizor natural comun, și anume 1.

Algoritmi de bază pentru parcurgerea divizorilor naturali ai unui număr natural nenul n

a) verificarea tuturor candidaților posibili din intervalul $[1, n]$

Pentru determinarea tuturor divizorilor naturali ai lui n pe parcurs toate numerele naturale din intervalul precizat, pentru fiecare verificându-se proprietatea cerută (dacă restul împărțirii lui n la numărul curent este 0).

De exemplu, pentru $n=100$, sunt necesari 100 de pași.

În limbajul C++, pentru determinarea tuturor divizorilor naturali ai lui n s-au urmat pașii precizați.

```
for(d=1;d<=n;d++)
if(n%d==0)
cout<<d<<' ';
```

b) verificarea tuturor candidaților posibili pentru a fi divizori proprii, din intervalul $[2, n/2]$

Pentru determinarea tuturor divizorilor proprii ai lui n pe parcurs toate numerele naturale din intervalul precizat, pentru fiecare verificându-se proprietatea cerută (dacă restul împărțirii lui n la numărul curent este 0). Dacă se dorește parcurgerea tuturor divizorilor, se au în vedere, în plus, 1 și n (dacă $n \neq 1$).

De exemplu, pentru $n=100$, sunt necesari **49** de pași. Dacă se dorește parcurgerea tuturor divizorilor, la cei determinați se adaugă 1 și n .

În limbajul C++, pentru determinarea tuturor divizorilor naturali ai lui n s-au urmat pașii precizați pentru obținerea divizorilor proprii, la care s-au adăgat 1 și n .	<pre>for(d=1;d<=n/2;d++) if(n%d==0) cout<<d<<' '; cout<<n<<' ';</pre>
--	--

c) verificarea tuturor candidaților posibili pentru a avea divizori-pereche, din intervalul $[1, \sqrt{n}]$

Dacă d este divizor al lui $n \Rightarrow n/d$ este divizor al lui n .

Se impune condiția $d < n/d$ ($d \cdot d < n$), pentru a nu găsi aceeași pereche, în ordine inversă.

Se tratează separat cazul $d=n/d$ ($d \cdot d = n$).

Pentru determinarea tuturor divizorilor lui n , se parcurg toate numerele naturale din intervalul precizat (determinându-se câte doi divizori la fiecare pas) și se tratează separat cazul $d \cdot d = n$.

Dacă se dorește determinarea divizorilor în ordine crescătoare, se parcurg toate numerele naturale din intervalul precizat, luându-se în considerare doar divizorul d determinat, apoi se tratează separat cazul $d \cdot d = n$ ($d = n/d$), și se reia parcurgerea în ordine descrescătoare a numerelor naturale din intervalul precizat, luându-se în considerare doar divizorul n/d , pentru fiecare divizor d determinat.

De exemplu, pentru $n=100$, sunt necesari **10** pași.

În limbaj C++, pentru determinarea tuturor divizorilor lui n , s-au parcurs pașii precizați.	<pre>for(d=1;d*d<n;d++) if(n%d==0) cout<<d<<' ' <<n/d<<' '; if(d*d==n) cout<<d<<' ';</pre>	Pentru afișarea divizorilor în ordine crescătoare: <pre>for(d=1;d*d<n;d++) if(n%d==0) cout<<d<<' '; if(d*d==n) cout<<d<<' '; for(d=d-1;d>=1;d--) if(n%d==0) cout<<n/d<<' ';</pre>
--	---	--

Algoritmi de bază pentru determinarea celui mai mare divizor comun a două numere naturale a și b

a) verificarea tuturor candidaților posibili din intervalul $[1,a] \cap [1,b]$

Pentru determinarea celui mai mare divizor comun, s-au parcurs toate numerele naturale din intervalul precizat și, pentru fiecare, s-a verificat dacă este divizor atât pentru a , cât și pentru b , actualizându-se la fiecare astfel de divizor comun variabila în care se memorează numărul căutat.

De exemplu, pentru $a=100$ și $b=330$ sunt necesari 100 de pași.

În limbaj C++, pentru determinarea celui mai mare divizor comun, s-au parcurs pașii precizați.

Numărul căutat este obținut în variabila cm .

```
for(d=1;d<=a && d<=b;d++)  
if(a%d==0 && b%d==0)  
cm=d;
```

b) algoritmul lui Euclid prin scăderi repetate

Dacă cm este cel mai mare divizor comun a numerelor a și b , atunci

$$a=cm \cdot a1$$

$$b=cm \cdot b1$$

$\Rightarrow a-b=cm \cdot (a1-b1) \Rightarrow cm$ este cel mai mare divizor comun a numerelor a , b , $a-b$, $b-a$.

Pentru determinarea celui mai mare divizor comun, algoritmul actualizează valoarea celui mai mare dintre cele două numere, prin scăderea din acesta a celuilalt număr; algoritmul reia acest pas în mod repetat, cât timp cele două numere sunt diferite, iar la final, valoarea egală obținută pentru acestea este cel mai mare divizor comun căutat.

De exemplu, pentru $a=100$ și $b=330$ sunt necesari 9 pași.

În limbaj C++, pentru determinarea celui mai mare divizor comun, s-au urmat etapele descrise.

Numărul căutat este obținut în variabila cm .

```
while(a!=b)  
if(a>b)  
a=a-b;  
else  
b=b-a;  
cm=a; //sau cm=b;
```

c) algoritmul lui Euclid prin împărțiri repetate

Pentru determinarea celui mai mare divizor comun, algoritmul determină, la fiecare pas, restul împărțirii primului număr (deîmpărțitul) la al doilea (împărțitorul), apoi actualizează valorile celor două numere, înlocuindu-se primul număr cu împărțitorul, iar al doilea număr cu restul

obținut; algoritmul reia acest pas în mod repetat, cât timp împărțitorul (al doilea număr) este nenul, iar la final, valoarea obținută pentru deîmpărțit (primul număr) este cel mai mare divizor comun căutat.

De exemplu, pentru $a=100$ și $b=330$ sunt necesari 5 pași.

În limbaj C++, pentru determinarea celui mai mare divizor comun, s-au urmat etapele descrise.

Numărul căutat este obținut în variabila cm .

```
while(b!=0)
{ r=a%b;
a=b;
b=r;
}
cm=a;
```

Algoritmi de bază pentru descompunerea unui număr în factori primi

Orice număr natural n se poate scrie în mod unic ca produs de factori primi, fiecare la o anumită putere: $n=d_1^{p_1} \cdot d_2^{p_2} \cdot \dots \cdot d_k^{p_k}$, unde d_1, d_2, \dots, d_k sunt numere prime (factori primi).

a) algoritmul prin eliminarea multiplilor unui divizor d

Se parcurg toate numerele naturale începând de la 2, cât timp sunt mai mici decât valoarea actualizată a lui n și, pentru fiecare, se determină puterea la care apare în descompunerea în factori primi a lui n , prin împărțiri exacte repetate ale lui n la numărul curent, actualizându-se la fiecare pas valoarea lui n , precum și puterea corespunzătoare; numerele pentru care puterea este nenulă sunt factori primi ai lui n .

De exemplu, pentru $n=100$, se verifică 5 numere.

În limbaj C++, pentru determinarea factorilor primi și a puterilor acestora, s-au urmat etapele descrise.

```
for(d=2;d<=n;d++)
{ p=0;
while(n%d==0)
{ n=n/d;
p++;
}
if(p!=0)
cout<<d<<'^'<<p<<'\n';
}
```

b) algoritmul prin eliminarea multiplilor unui divizor d , îmbunătățit

Dacă $d*d > n \Rightarrow d$, dacă este divizor, este chiar egal cu n , el fiind ultimul divizor posibil care ar trebui testat.

De exemplu, pentru $n=3400$, se verifică 7 numere.

În limbaj C++, pentru determinarea celui mai mare divizor comun, s-au urmat etapele descrise. Factorii primi determinați și puterile acestora sunt afișați pe ecran.

```
for(d=2;d*d<=n;d++)
{ p=0;
  while(n%d==0)
  { n=n/d;
    p++;
  }
  if(p!=0)
  cout<<d<<"^"<<p<<endl;
}
if(n>1)
  cout<<n<<" "<<1;
```

Probleme recomandate

Să se scrie câte un program/subprogram C/C++ prin care să se rezolve cerințele de mai jos.

A. Se citește de la tastatură un număr natural n ($n \in [1, 10^5]$).

1. Să se afișeze pe ecran toți divizorii naturali ai lui n
 - a) în ordine crescătoare;
 - b) în ordine descrescătoare.
2. Să se afișeze pe ecran toți divizorii proprii ai lui n .
3. Să se afișeze pe ecran numărul divizorilor naturali ai lui n .
4. Să se verifice dacă n este prim, afișând pe ecran un mesaj corespunzător.
 - a) utilizând pentru implementare numărarea divizorilor;
 - b) utilizând pentru implementare numărarea divizorilor proprii;
 - c) utilizând pentru implementare o variabilă de tip semafor.
5. Să se afișeze pe ecran suma tuturor divizorilor lui n .
6. Să se afișeze pe ecran produsul divizorilor primi ai lui n .

B.

1. Să se afișeze pe ecran cel mai mare divizor comun a trei numere naturale din intervalul $[1, 10^5]$, citite de la tastatură.
 - a) utilizând pentru implementare verificarea tuturor divizorilor posibili;
 - b) utilizând pentru implementare metoda lui Euclid cu scăderi repetate;
 - c) utilizând pentru implementare metoda lui Euclid cu împărțiri repetate;
2. Să se afișeze pe ecran cel mai mare divizor comun a n ($n \in [1, 100]$) numere din intervalul $[1, 10^5]$, citite de la tastatură.
3. Să se afișeze factorul prim care apare la puterea maximă în descompună în factori primi a numărului natural n ($n \in [1, 10^5]$), citit de la tastatură.

C.

1. Se citesc de la tastatură două numere naturale a și b din intervalul $[1, 10^5]$, cu $a < b$. Să se afișeze pe ecran toate numerele prime din intervalul $[a, b]$.
2. Se citesc de la tastatură două numere naturale a și b din intervalul $[1, 10^5]$, cu $a < b$. Să se afișeze pe ecran numărul de valori prime din intervalul $[a, b]$.
3. Se citește de la tastatură un număr natural x din intervalul $[1, 10^5]$. Să se afișeze pe ecran toate numerele prime mai mici decât x .
4. Se citește de la tastatură un număr natural x din intervalul $[1, 10^5]$. Să se afișeze pe ecran primul număr prim mai mare decât x .
5. Se citește de la tastatură un număr natural x din intervalul $[1, 10^5]$. Să se afișeze pe ecran ultimul număr prim mai mic decât x .
6. Se citește de la tastatură un număr natural k din intervalul $[1, 10^5]$. Să se afișeze pe ecran primele k numere prime.
7. Se citesc de la tastatură două numere naturale a și b din intervalul $[1, 10^5]$, cu $a < b$. Să se verifice dacă intervalul $[a, b]$ conține cel puțin un număr impar prim.
8. Se citesc de la tastatură două numere naturale a și b din intervalul $[1, 10^5]$, cu $a < b$. Să se verifice dacă în intervalul $[a, b]$ toate numerele impare sunt prime.

II.2.2. Teme pentru recapitulare – șiruri de caractere

Tematica șirurilor de caractere apare în multe dintre subiectele de la examenul național de bacalaureat, la disciplina informatică, specializarea matematică-informatică (și intensiv informatică) respectiv la clasele din filiera vocațională, profil militar, specializarea matematică-informatică.

Pentru pregătirea acestuia, în limbajul C++ se recomandă recapitularea următoarelor aspecte legate de șirurile de caractere:

- constante de tip șir de caractere;
- variabile care permit memorarea unui șir de caractere și a unei liste de șiruri de caractere;
- operații specifice pentru citire, afișare, acces la un caracter al unui șir, atribuire, adunarea unui întreg;
- funcții predefinite pentru lungimea unui șir de caractere, copierea, concatenarea, căutarea, compararea, separarea unui șir în subșiruri, transformări din șiruri în numere și invers.

Constante de tip șir de caractere

Exemple:

"acesta este un sir!"

"123"

"78^%\$#"

"" (șirul vid)

Observație: 'a' (un singur caracter) este diferit de "a" (șir format din două caractere: caracterul util 'a', urmat de caracterul de sfârșit de șir).

Observație: între o literă mică și litera mare corespunzătoare există relația:

literaMica-literaMare='a'-'A'

Variabile care permit memorarea șirurilor de caractere

Declararea unei variabile care permite memorarea unui șir de caractere

a) cu alocare de memorie (tablou unidimensional cu elemente de tip char)

```
char s[dimmaxima+1];
```

Variabila s memorează adresa de la care începe zona de memorie rezervată pentru șir.

b) cu alocare de memorie (tablou unidimensional cu elemente de tip char) și inițializare

```
char s[dimmaxima+1]=constanta;
```

Exemplu: `char s[10]="initial";`

Caracterul `'\0'` este caracterul cu codul 0 și marchează sfârșitul șirului.

c) fără alocare de memorie

```
char *s;
```

Se poate utiliza, de exemplu, când șirul este deja memorat printr-o altă variabilă și doar se prelucrează cu ajutorul lui `s`, declarat astfel.

Declararea unei variabile care permite memorarea unei liste de șiruri de caractere

Mai multe șiruri de caractere se pot memora ca un tablou unidimensional de șiruri de caractere (adică un tablou de tablouri de caractere), deci un tablou bidimensional, în care fiecare linie corespunde câte unui șir.

```
char lista[nrMaxSiruri][nrMaxCaractereInSir+1];
```

`lista[i]` este șirul aflat pe linia `i` a tabloului bidimensional.

Operații specifice

Citirea de la tastatură a unui șir de caractere

a) citirea unui cuvânt (șir de caractere care nu conține spații)

```
cin>>s;
```

Citirea sare peste caracterele albe (blanc, tab, enter) de la început, citește șirul, apoi se oprește la primul caracter alb.

b) citirea unei fraze (un șir care conține sau nu spații).

```
cin.get(s,dimMax+1,CarFinal);
```

Citirea începe de la poziția curentă (chiar dacă pe acea poziție este un caracter alb - blanc, tab, enter), și se încheie după ce s-au citit `dimMax` caractere (inclusiv caracterele albe) sau dacă printre cele `dimMax` caractere s-a întâlnit `carFinal`. Dacă nu se specifică un `carFinal`, implicit el se consideră Enter.

c) citirea mai multor fraze (mai multe șiruri, scrise pe câte o linie, fiecare conținând sau nu spații)

```
cin.get(s,dimMaxima+1,CarFinal);
```

```
cin.get(); //sare peste caracterul final
```

d) citirea unei linii (șir de caractere scris pe o linie, care conține sau nu spații)

```
cin.getline(s,dimMax+1);
```

Citirea începe de la poziția curentă (chiar dacă pe acea poziție sunt caractere albe - blanc,

tab, enter), și se încheie după ce s-au citit dimMax caractere (inclusiv caracterele albe) sau dacă printre cele dimMax caractere s-a întâlnit Enter (caz în care se ignoră celelalte caractere de pe linia respectivă).

Afișarea pe ecran a unui șir de caractere

```
cout<<s;
```

Se afișează toate caracterele, începând de la adresa s, până la întâlnirea caracterului \0.

Accesul la un caracter al șirului

s[i] este caracterul de pe poziția i (prima poziție este 0).

Dacă s[i]=='\0' (sau dacă s[i]==0), înseamnă că s-a încheiat șirul.

Atribuirii

Fie două șiruri, declarate cu alocare de memorie, s1, s2, și două șiruri declarate fără alocare de memorie s3 și s4. De exemplu:

```
char s1[10], s2[15], *s3, *s4;
```

Tipuri de atribuirii care <i>SUNT</i> permise	Tipuri de atribuirii care <i>NU SUNT</i> permise
s1[3]='*'; //sau orice constantă de tip caracter	s1="abcd"; //se permite numai la declarare
s1[1]=s2[3];	s3="abcd"; //se permite numai la declarare
s3=s1;	s1=s3;
s3=s4;	s1=s2;

Adunarea unui întreg

s+nr

Este adresa de memorie aflată cu nr octeți (caractere) după adresa s, deci este adresa caracterului de pe poziția nr (s[nr]).

Funcții predefinite specifice

Pentru utilizarea funcțiilor predefinite, se include biblioteca specifică:

```
#include <cstring> //specific C++
```

```
#include <string.h> //specific C
```

Elevii pot utiliza oricare dintre funcțiile care prelucrează șirurile de caractere, pentru rezolvarea subiectelor de la examenul național de bacalaureat fiind suficiente cele prezentate în manualele în vigoare.

Lungimea unui șir – strlen

strlen(s)=>nr de caractere utile ale unui șir

Copierea unui șir – strcpy strncpy

strcpy(destinatie,sursa)=>copiază șirul aflat la adresa sursa, la adresa destinatie și returnează adresa șirului obținut (la adresa destinatie).

strncpy(destinatie,sursa,n)=>copiază primele n caractere începând de la adresa sursa, la adresa destinatie și returnează adresa șirului obținut (la adresa destinatie)

Observație: caracterul \0 se copiază numai dacă este cuprins între cele n caractere.

Observație: funcția se poate utiliza pentru inserarea/eliminarea unui subșir dintr-un șir.

Concatenarea a două șiruri – strcat strncat

strcat(destinatie,sursa)=>lipește șirul aflat la adresa sursa, la finalul șirului aflat la adresa destinatie și returnează adresa șirului obținut (la adresa destinatie)

strncat(destinatie,sursa,n)=>lipește primele n caractere începând de la adresa sursa, la finalul șirului aflat la adresa destinatie și returnează adresa șirului obținut (la adresa destinatie)

Observație: caracterul \0 se copiază numai dacă este cuprins între cele n caractere.

Căutarea unui subșir într-un șir – strchr strstr

strchr(sir,caracter)=>caută caracterul în sir și returnează ADRESA (nu poziția) la care a găsit prima apariție a acestuia, sau adresa nulă dacă nu a găsit caracterul în sir (0 sau NULL)

strstr(sir,subsir)=>caută subșirul în sir și returnează ADRESA (nu poziția) la care a găsit prima apariție a acestuia, sau adresa nulă dacă nu a găsit subșirul în sir (0 sau NULL sau null)

Compararea (lexicografică – ca în dicționar/catalog) a două subșiruri – strcmp

strcmp(sir1,sir2)=>

>0, dacă sir1>sir2 (sir1 se află în dicționar după sir2)

=0, dacă sir1=sir2

<0, dacă sir1<sir2

Separarea unui șir în subșiruri (de ex fraza în cuvinte) - strtok

a) primul apel

strtok(sir,sirSeparatori)=>adresa primului subșir (cuvânt). Inserează după primul subșir un caracter de final=>strică șirul inițial

b) următoarele apeluri

strtok(NULL,sirSeparatori)=>adresa următorului subșir (cuvânt). Începe căutarea din poziția în care a inserat NULL ultima oară și identifică următorul cuvânt, inserând după

acesta un nou caracter NULL

De exemplu, secvența următoare afișează toate cuvintele unei fraze accesate prin șirul s, utilizând separatorii spațiu, virgulă și punct și virgulă.

```
for(char *cuv= strtok(s, " ,;"); cuv!=NULL; cuv= strtok(NULL, " ,;"))
    cout<<cuv<<"\n";
```

Transformare a unui șir în număr

atoi(sir)=>returnează numărul ale cărui cifre sunt memorate în sir

itoa(nr, sir, baza)=> în sir se află șirul cifrelor numărului nr

Observație: există funcții similare pentru numere reale (atof, ftoa etc)

Observație: transformarea unei cifre (simbol) în valoarea corespunzătoare – diferența față de cifra reper '0'

cifraSimbol-'0'=>valoarea numerică a cifrei

Probleme propuse

A.

1. Se citește o frază cu cel mult 100 de caractere, numai litere mici și spații. Să se afișeze numărul vocalelor sale.
2. Se citește o frază cu cel mult 100 de caractere, litere mici și mari și spații. Să se afișeze fraza în care s-au înlocuit literele mici cu litere mari.
3. Să se afișeze numărul de apariții ale fiecărei litere a unui text. Textul conține numai litere mici ale alfabetului englez.

B.

4. Se citesc n cuvinte, scrise unul sub altul. Să se afișeze toate cuvintele care au aceeași lungime cu ultimul citit.
5. Se citesc n cuvinte, scrise unul sub altul. Să se afișeze cuvintele în ordine alfabetică.

C.

6. Se citește o frază cu cel mult 100 de caractere, litere mici și mari și spații. Să se înlocuiască în memorie toate aparițiile unui cuvânt dat, x, cu alt cuvânt dat, y.
7. Se citește o frază cu cel mult 100 de caractere, litere mici și mari și spații. Să se elimine în memorie toate aparițiile unui cuvânt dat, x.
8. Se citește o frază cu cel mult 100 de caractere, litere mici și mari și spații. Să se insereze în memorie cuvântul dat x după toate aparițiile unui alt cuvânt dat, y.

II.2.3. Teme pentru recapitulare – Metoda backtracking

Metoda backtracking apare în multe dintre subiectele de la examenul național de bacalaureat, la disciplina informatică, specializarea matematică-informatică (și intensiv informatică), respectiv la clasele din filiera vocațională, profil militar, specializarea matematică-informatică.

Pentru pregătirea acestuia, se recomandă recapitularea următoarelor aspecte legate de metoda backtracking:

- structurarea datelor cu ajutorul tablourilor/șirurilor de valori;
- pașii algoritmului și repere pentru rezolvarea problemelor cu metoda backtracking;
- identificarea caracteristicilor metodei pentru generarea permutărilor unei mulțimi (ordine, valori distincte, număr complet de elemente), aranjamentelor (ordine, valori distincte, număr parțial de elemente), combinărilor (nu contează ordinea/ordine impusă, valori distincte, număr parțial de elemente), submulțimilor (nu contează ordinea/ordine impusă, valori distincte, număr variabil de elemente), produs cartezian (ordine, valori repetabile, număr constant de elemente), partiții ale unei mulțimi (repartizare a tuturor elementelor), sume (transmiterea sumei parțiale).
- aplicarea metodei pentru a identifica o soluție care urmează/precede unei soluții date, ultima/penultima soluție generată etc.

Prezentarea metodei

Metoda de programare backtracking (iterativă sau recursivă):

- este o metodă de parcurgere sistematică a spațiului soluțiilor posibile ale unei probleme;
- este o metodă generală de programare care se poate adapta și se poate aplica tuturor problemelor ale căror soluții se pot reprezenta prin câte un tablou
 $v[]=(v[1],v[2],v[3],v[4],\dots,v[n]);$
- pentru a putea aplica metoda, mulțimea valorilor posibile ale fiecărui element al tabloului-soluție trebuie să fie finită și ordonată;
- cu ajutorul acestei metode se pot obține toate soluțiile posibile ale unei probleme;
- este cea mai costisitoare metodă din punctul de vedere al timpului de executare.

Generarea soluțiilor problemei cu metoda backtracking

Se memorează o soluție într-un tablou unidimensional

$$v[] = (v[1], v[2], v[3], v[4], \dots, v[n])$$

în care fiecare element $v[i]$ aparține unei mulțimi finite A_i ale cărei elemente se află într-o relație de ordine. Aceste mulțimi A_i pot fi identice.

$$v[i] \in A_i$$

Reprezentarea tabloului v și a mulțimilor A_1, A_2, \dots, A_n

$V[n]$	<table border="1"><tr><td> </td></tr></table>		$A_n = \{a_{n1}, a_{n2}, a_{n3}, \dots, a_{tn}\}$
.....	
$V[i]$	<table border="1"><tr><td> </td></tr></table>		$A_i = \{a_{i1}, a_{i2}, a_{i3}, \dots, a_{ti}\}$
.....	
$V[3]$	<table border="1"><tr><td> </td></tr></table>		$A_2 = \{a_{31}, a_{32}, a_{33}, \dots, a_{t3}\}$
$V[2]$	<table border="1"><tr><td> </td></tr></table>		$A_2 = \{a_{21}, a_{22}, a_{23}, \dots, a_{t2}\}$
$V[1]$	<table border="1"><tr><td> </td></tr></table>		$A_1 = \{a_{11}, a_{12}, a_{13}, \dots, a_{t1}\}$

Algoritmul de generarea a soluțiilor unei probleme cu metoda backtracking

Algoritmul de tip backtracking construiește vectorul $v[]$ (numit vector soluție) astfel:

Se începe de pe prima poziție ($i=1$) și se completează elementele acestui tablou pe rând, astfel încât pe poziția i , curentă, se testează toate valorile posibile pentru această poziție, iar la stabilirea valorii corecte $v[i]$, se ține cont de valorile deja selectate pentru pozițiile anterioare ($v[1], v[2], \dots, v[k-1]$), care se știe că sunt corecte (și deci nu se mai verifică din nou corectitudinea acestora).

Presupunem ca s-a generat deja prefixul $v[1], v[2], \dots, v[i-1]$, care se știe că respectă condițiile impuse, iar poziția curentă este i .

Pasul 0: dacă prefixul $v[1], v[2], \dots, v[i-1]$ este o soluție completă, acesta se afișează/prelucrează, și se revine la poziția anterioară, $i-1$, reluându-se pentru această poziție ($i-1$) pasul 1 (știind că deja au fost testate o parte dintre valori pentru această poziție la care s-a revenit).

Pasul 1: se alege pentru poziția curentă, i , o valoare posibilă pentru $v[i]$, din mulțimea A_i , dintre cele care nu au fost încă testate pentru acest prefix. Se verifică dacă elementul ales îndeplinește condițiile/restricțiile problemei, știind că elementele $v[1], v[2], \dots, v[i-1]$, au valori corespunzătoare și că elementele $v[i+1], \dots, v[n]$ nu au, încă, valori atribuite.

- dacă valoarea aleasă pentru $v[i]$ este o valoare validă (ce respectă criteriile impuse/nu contrazice criteriile impuse), se trece la poziția următoare, $i+1$, și se reia pasul 1 pentru această poziție ($i+1$);
- dacă valoarea $v[i]$ nu este validă, se rămâne pe poziția i , și se reia pasul 1.

Pasul 2: dacă pentru poziția i curentă nu mai există nicio valoare din A_i netestată, se revine la poziția $i-1$, și se reia pasul 1 pentru această poziție (știind că deja au fost testate o parte dintre valori pentru această poziție la care s-a revenit).

Algoritmul se încheie dacă pentru **prima poziție** s-au testat TOATE valorile posibile.

Repere pentru rezolvare

A. Ce este o soluție?

Ce reprezintă elementul $v[i]$ în cadrul soluției?

B. Care este mulțimea valorilor posibile pentru $v[i]$?

-trebuie să fie finită;

-trebuie să fie ordonată – dacă nu este ordonată, se numerotează elementele:

$v[i] \in \{p, p+1, \dots, u\}$ //elemente consecutive

C. Care sunt condițiile de validare?

Știind că $v[1], v[2], \dots, v[i-1]$ reprezintă un prefix al soluției, prefix ce respectă condițiile impuse, se cere să se verifice dacă $v[i]$, adăugat prefixului, corespunde cerinței.

D. În ce condiții se afișează o soluție?

(Obs. Dacă numărul de elemente ale soluției este constant, soluția se afișează dacă s-au completat toate elementele, deci dacă $i = \text{nr. de elemente} + 1$ – vezi pasul 0).

Probleme propuse:

1. Utilizând metoda backtracking se generează toate șirurile de 5 litere distincte din mulțimea $\{A, M, U, R, G\}$, astfel încât în fiecare șir litera R precede literele A și M. Primele cinci soluții generate sunt, în această ordine: URAMG, URAGM, URMAG, URMGA, URGAM. Scrieți soluția generată imediat după GURMA.
2. Utilizând metoda backtracking, se generează toate amestecurile de apă provenită din surse distincte din mulțimea $\{\text{lac, mare, ocean, ploaie, râu}\}$, astfel încât o sursă să fie de apă sărată și una sau două surse să fie de apă dulce. Marea și oceanul sunt surse de apă sărată, iar lacul, ploaia și râul sunt surse de apă dulce. Două amestecuri sunt distincte dacă diferă prin cel puțin o sursă a apei. Primele patru soluții obținute sunt, în această ordine: (lac, mare), (lac, mare, ploaie), (lac, mare, râu) și (lac, ocean). Scrieți soluția generată imediat înainte și soluția generată imediat după (ocean, ploaie).

3. Utilizând metoda backtracking, se generează toate parfumurile formate prin amestecarea a câte 3 esențe distincte din mulțimea {bergamotă, cireș, iris, lămâie, salcâm}. Două parfumuri sunt distincte dacă diferă prin cel puțin o esență. Primele patru soluții obținute sunt, în această ordine: (bergamotă, cireș, iris), (bergamotă, cireș, lămâie), (bergamotă, cireș, salcâm) și (bergamotă, iris, lămâie). Scrieți penultima soluție generată.
4. Utilizând metoda backtracking, se generează toate pachetele formate din câte 3 tipuri distincte de ceai din mulțimea {matcha, mate, moringa, oolong, tulsi}. Două amestecuri sunt distincte dacă diferă prin cel puțin un tip de ceai. Primele patru soluții obținute sunt, în această ordine: (matcha, mate, moringa), (matcha, mate, oolong), (matcha, mate, tulsi) și (matcha, moringa, oolong). Indicați succesiunea care NU se obține, prin această metodă, în ordinea dată.
 - a. (matcha, moringa, tulsi)
 - b. (moringa, oolong, tulsi)
 - c. (mate, moringa, oolong)
 - d. (oolong, tulsi, mate)

II.2.4. Teme pentru recapitulare – Grafuri

Pentru tematica referitoare la grafuri și arbori, la recapitularea în vederea pregătirii examenului de bacalaureat, se vor avea în vedere:

- Definiții specifice grafurilor neorientate și orientate;
- Terminologie
- Tipuri de grafuri și grafuri asociate
- Reprezentare în memorie

Grafuri

Definiții specifice

Graf = o mulțime finită V care are asociată o relație binară internă L . Graful se va nota $G=(V,L)$.
 G – graful, V – mulțimea vârfurilor/nodurilor, L – mulțimea legăturilor între vârfuri, adică mulțimea arcelor/muchiilor.

Graf neorientat = un graf în care mulțimea legăturilor L este **simetrică** - dacă elementul (p, q) aparține L atunci și elementul (q, p) aparține L .

V – mulțime a nodurilor; L – mulțime a muchiiilor

Graf orientat = un graf în care mulțimea legăturilor L nu este simetrică.

V – mulțime a vârfurilor; L – mulțime a arcelor.

Terminologie

Grafuri neorientate

Graf complet = graf neorientat în care există muchie între oricare două noduri.

Observație: numărul de muchii într-un graf complet se calculează cu formula:

$$\frac{n \cdot (n - 1)}{2}$$

unde n este numărul de noduri.

Graf conex = graf în care între oricare două noduri există cel puțin un lanț cu extremități în acestea.

Componentă conexă = subgraf al grafului de referință maximal în raport cu proprietatea de conexitate.

Lanț hamiltonian = lanț elementar care conține toate nodurile unui graf.

Ciclu hamiltonian = ciclu elementar care conține toate nodurile unui graf.

Graf hamiltonian = graf care conține cel puțin un ciclu hamiltonian.

Condiție suficientă pentru un graf cu cel puțin trei noduri: dacă gradul oricărui nod este mai mare decât $n/2$ atunci graful este hamiltonian.

Lanț eulerian = lanț simplu care conține toate muchiile unui graf.

Ciclu eulerian = ciclu care conține toate muchiile unui graf.

Grafuri orientate

Turnir = graf orientat în care există un singur arc între oricare două vârfuri.

Graf complet = graf orientat în care oricare două vârfuri distincte ale sale sunt adiacente (pentru două vârfuri x și y există arcul (x,y) și/sau există arcul (y,x)).

Numărul de grafuri orientate complete cu n noduri

$$\text{este } 3^{\frac{n \cdot (n-1)}{2}}.$$

Graf tare conex = graf în care între oricare două vârfuri x, y există cel puțin un drum cu extremitatea inițială în x și extremitatea finală în y , și cel puțin un drum cu extremitatea inițială în y și extremitatea finală în x .

Componentă tare conexă = subgraf al grafului de referință maximal în raport cu proprietatea de tare conexitate.

Drum hamiltonian = drum elementar care conține toate vârfurile unui graf.

Circuit hamiltonian = circuit elementar care conține toate vârfurile unui graf.

Graf hamiltonian = graf care conține cel puțin un circuit hamiltonian.

Drum eulerian = drum simplu care conține toate arcele unui graf.

Circuit eulerian = circuit care conține toate arcele unui graf.

Graf eulerian = graf care conține cel puțin un ciclu eulerian.

Teoremă: Un graf neorientat conex este eulerian dacă și numai dacă gradul oricărui nod este par.

Graf eulerian = graf care conține cel puțin un circuit eulerian.

Teoremă: Un graf orientat este eulerian dacă și numai dacă pentru orice vârf gradul interior este egal cu gradul exterior.

Metode de reprezentare a grafurilor

Graf neorientat

Matrice de adiacență pentru $G = (V, L)$

$A[i][j] = 1$ dacă există muchia $[i, j]$ în mulțimea L , și este 0, altfel

Observație 1: diagonala principală conține numai valori de 0.

Observație 2: matricea este simetrică față de diagonala principală.

Liste de adiacență

Listele de adiacență memorează pentru fiecare nod vecinii lui (nodurile adiacente cu el).

Graf orientat

Matrice de adiacență

$A[i][j] = 1$ dacă există arcul (i, j) în mulțimea L , și este 0, altfel

Observație 1: diagonala principală conține numai valori de 0.

Liste de adiacență - succesori

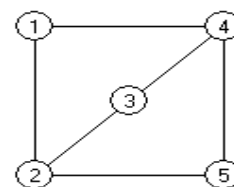
Listele de succesori memorează pentru fiecare nod i vecinii lui, j , cu proprietatea că există arcul (i, j) .

Liste de adiacență - predecesori

Listele de predecesori memorează pentru fiecare nod i vecinii lui, j , cu proprietatea că există arcul (j, i) .

Probleme propuse:

1. Precizați numărul minim de noduri pe care îl poate conține un graf neorientat cu 50 de muchii, și în care 15 noduri sunt izolate.
2. Se consideră un graf neorientat cu nodurile: 1, 2, 3, 4, 5, 6, 7, 8 și muchiile: [1,3], [1,7], [2,6], [3,7], [5,2], [5,6], [8,4]. Precizați numărul de componente conexe ale grafului.
3. Se consideră un graf neorientat cu nodurile: 1,2,3,4,5,6,7,8 și muchiile: [1,3], [1,7], [2,6], [3,7], [5,2], [5,6], [8,4]. Precizați numărul minim de muchii ce pot fi adăugate astfel încât graful să devină conex.
4. Precizați numărul maxim de vârfuri izolate pe care le poate avea un graf neorientat cu 8 noduri și 12 muchii.
5. Se consideră graful neorientat din figura alăturată. Precizați numărul maxim de muchii ce pot fi eliminate din graf astfel încât în graful parțial rezultat să fie conex .
6. Se consideră graful orientat cu 5 noduri numerotate de la 1 la 5 și cu arcele: (1,2), (2,1), (2,5), (3,2), (4,3), (5,1), (5,2), (5,4). Determinați gradul intern al nodului cu gradul extern maxim.



II.3. Evaluarea în mediul on-line a performanțelor școlare

Evaluarea în mediul on-line a performanțelor școlare și a competențelor elevilor este reglementată de Metodologia-cadru de evaluare în mediul on-line a performanțelor școlare și a competențelor elevilor, aprobată prin O.M.E. nr. 3750/2023.

Conform Metodologiei, evaluarea în mediul on-line a performanțelor școlare și a competențelor elevilor va respecta următoarele principii: a) claritate și transparență; b) validitate; c) diversitate; d) flexibilitate; e) fiabilitate; f) accesibilitate; g) implicare; h) ritmicitate; i) obiectivitate; k) echitate; l) securitate și confidențialitate.

II.3.1. Recomandări privind evaluarea în mediul on-line a performanțelor școlare și a competențelor școlare

Evaluarea asistată de mijloace IT este utilizată pentru a evalua competențele elevilor și include toate tipurile de evaluări, fie că este vorba de evaluare formativă sau sumativă. Livrarea testelor se face atât on-line, cât și offline.

Evaluarea asistată de calculator urmărește evaluarea formativă în care elevii sunt ajutați să primească un feedback automat, legat de competențele proprii.

Testarea adaptivă pe calculator este o dezvoltare recentă a tehnologiei, care îmbunătățește în mare măsură evaluarea prin facilitățile oferite. În timp ce un elev răspunde la întrebări, tehnologia este capabilă să ajusteze nivelul de dificultate, în funcție de succesul sau greșelile acestuia în rezolvarea sarcinilor de lucru.

Portofoliul electronic este un proces de colectare a probelor electronice, menținute pe web. Astfel de probe electronice includ fișiere text, produse multimedia, imagini și hyperlinkuri.

Modul de abordare a evaluării la disciplina informatică trebuie să aibă în vedere complexitatea disciplinei, care încurajează și dezvoltă gândirea critică, dar și aplicarea în practică a celor învățate.

II.3.2. Exemple de platforme/aplicații care sprijină elaborarea și administrarea de instrumente de evaluare

Oracle Academy – este o platformă care pune la dispoziție curriculum specific bazelor de date Oracle, și permite elevilor să învețe și să exerseze în Oracle Autonomous Database prin Oracle APEX și programul Oracle Academy Cloud. Profesorii au la dispoziție elemente pentru gestionarea claselor de elevi, urmărirea progresului elevilor, administrarea de probe de evaluare.

W3schools – este o platformă care oferă o gamă largă de servicii și produse pentru începători și profesioniști, oferind o gamă largă de tutoriale, dintre care evidențiem pe cele corespunzătoare programei clasei a XII-a: HTML, CSS, MySQL, JavaScript, SQL, PHP, C# etc.

DigitalEdu - Resurse Educaționale Deschise – este o bază de date cu resurse educaționale în format digital, create de cadre didactice, organizate pe discipline și ani de studiu. Conține link-uri care permit accesul la peste 7000 de teste, fișe de lucru, jocuri didactice, filme și simulări.

Kinderpedia - Soluție completă de comunicare și management pentru unități de învățământ, disponibilă pe web, dar și ca aplicație pentru dispozitive mobile (pentru sistemele de operare Android și iOS). Platforma este accesibilă gratuit și ajută la gestionarea clasei/grupeii și a activităților de învățare, dar și la comunicare și colaborare. Este adaptată pentru învățământul preuniversitar și este în limba română. Are modul de videoconferință (Zoom) integrat cu orarul clasei, dar și catalog electronic cu posibilitatea de atribuire de teme către elevi și de notare a rezolvărilor acestora. Elevii se pot înscrie doar cu un cod unic furnizat de profesor/învățător (fără a avea nevoie de adrese de email). Sunt disponibile mai multe tutoriale pentru utilizarea platformei, în limba română: <https://docs.kinderpedia.co/ro/>

WAND.education - Permite crearea de la zero a lecțiilor (oferă multe modele și șabloane de lecții) sau preluarea unor lecții create de alți profesori. Este simplu de folosit în crearea de conținut educațional interactiv, lecția și/sau testul putând fi transmise în timp real elevilor; iar aceștia putându-le vizualiza pe orice tip de dispozitiv (desktop, laptop sau telefon mobil) bazat pe: iOS, Android, Mac OS, Windows. Are funcții de notare, evaluare și urmărire a parcursului școlar al elevilor.

Zoom & Google Classroom (video) - Există tutoriale video pentru instalarea și utilizarea Zoom și Google Classroom.

Digitaliada - Este o platformă cu resurse educaționale digitale create de cadre didactice, care conține exerciții, teste, tutoriale video și ghiduri. Secțiunea *Învățare și testare on-line* este structurată pe patru niveluri de acces, fiind dedicată atât directorilor, cadrelor didactice, indiferent de materia predată, cât și părinților și elevilor.

Cu **Microsoft Forms** și **Google Forms**, se pot crea sondaje, teste și anchete și se pot vedea cu ușurință rezultatele în timp real.

Din septembrie 2020, s-a lansat live platforma **Khan Academy** în limba română, care se poate accesa la adresa: <https://ro.khanacademy.org/>. **Khan Academy** este o organizație non-profit, având misiunea de a furniza gratuit educație la nivel mondial. Pe platforma organizației se pot crea clase și chiar importa clase de elevi din Google Classroom. Pentru disciplina informatică recomandăm lecțiile de programare postate pe platformă. Evaluarea se realizează cu ajutorul proiectului.

LIVRESQ (www.livresq.com) este un editor de resurse educaționale în format digital. Acesta facilitează crearea de lecții interactive, ce conțin texte, galerii de poze, animații, conținut audio, video, quiz-uri și alte elemente, fără a fi necesare cunoștințe de programare.

PBINFO (www.pbinfo.ro) este o platformă care permite rezolvarea de itemi de tip obiectiv (întrebări cu răspuns multiplu), semiobiectiv (întrebări cu răspuns scurt) sau subiectiv (rezolvarea de probleme, prin scrierea de programe în limbaj de programare), cu evaluare automată a răspunsurilor. De asemenea, permite construirea unor teste, teme etc.

INFOARENA (<https://infoarena.ro>) este o platformă care permite rezolvarea de itemi de tip subiectiv (de tip rezolvare de probleme, prin scrierea de programe în limbaj de programare), cu evaluare automată a răspunsurilor.

II.4. Abordări multi-, inter- și transdisciplinare

Învățarea interdisciplinară încurajează elevii să combine cunoștințe, idei, deprinderi de lucru din diferite discipline de studiu atunci când lucrează la un proiect sau rezolvă anumite probleme. Este adesea centrată pe învățarea activă, bazată pe probleme, pe găsirea de noi soluții inovatoare și punerea lor în practică.

Învățarea interdisciplinară extinde experiența de învățare și promovează gândirea critică prin încurajarea elevilor să se angajeze în rezolvarea de probleme complexe.

Profesorii care doresc să ofere studenților lor acest tip de educație trebuie:

- să aibă suficiente cunoștințe în fiecare disciplină pe care intenționează să le abordeze în dezvoltarea aplicațiilor interdisciplinare sau să colaboreze cu profesori de la alte discipline;
- să găsească echilibrul potrivit și nivelul de complexitate pentru sarcinile de lucru pe care le propune spre rezolvare,
- să convingă elevii că educația interdisciplinară este o activitate importantă;
- să rămână obiectiv și deschis la alte opinii și puncte de vedere;
- să promoveze sinteza ideilor din mai multe discipline;
- să gestioneze eficient clasa pentru a menține elevii calmi, organizați și concentrați asupra sarcinilor de lucru propuse.

În mod colectiv, provocarea educației interdisciplinare constă în faptul că necesită ca elevii și profesorii să investească timp și efort suplimentar, rămânând în același timp deschiși cu privire la modelul de învățare.

Temele contemporane care se pliază pe abordări multi-, inter- și trans- disciplinare pot fi inspirate din:

- educația privind mediul și schimbările climatice (inclusiv în relație cu programul Săptămâna verde);
- educația pentru siguranță și securitate (împotriva calamităților, cibernetică);
- educația privind mass-media (inclusiv abordarea critică a surselor de informare);
- educația financiară;
- educația juridică;
- educația antreprenorială;
- educația tehnologică;
- educația rutieră;
- educația civică;
- educația interculturală;
- educația pentru cetățenie democratică;
- educația pentru egalitate de șanse.

Majoritatea domeniilor prezentate anterior lucrează cu un volum mare de date care trebuie modelate, implementate și gestionate ulterior și care sunt supuse interogării pentru a răspunde unor alte provocări.

Proiectarea unei cercetări interdisciplinare începe cu „designul conceptual” care abordează „de ce” și „ce” al unui proiect de cercetare la nivel conceptual pentru a stabili obiectivele comune esențiale pentru colaborarea interdisciplinară. Designul conceptual include în principal activități precum gândirea, schimbul de cunoștințe interdisciplinare, citirea și discutarea. Produsul designului conceptual se numește „cadru conceptual” care cuprinde obiectivul cercetării, teoria sau teoriile care sunt centrale în proiectul de cercetare, întrebările de cercetare și operaționalizarea conceptelor care sunt măsurate sau înregistrate în timpul observării fenomenelor sau proceselor modelate.

Apoi, trebuie luate decizii cu privire la modul în care trebuie colectate datele, prin intermediul instrumentelor certificate, observații, interviuri, chestionare, interogări asupra bazelor de date existente sau o combinație a acestora.

Programarea vizuală studiată la modulul corespunzător programei clasei a XII-a, permite, prin aplicațiile vizuale create, abordarea unor probleme interdisciplinare cum ar fi:

- realizarea de animații care să simuleze fenomene fizice, reacții chimice;
- probleme de logica matematică;
- realizarea de jocuri
- crearea și evaluarea unor teste de cultură generală sau specifice unor domenii.

În cazul studierii modulului Programare WEB, se pot aborda teme teoretice și practice specifice oricărei alte discipline ca subiecte pentru realizarea unor pagini web.

Competițiile școlare propuse de Ministerul Educației permit valorificarea interdisciplinarității:

- Olimpiada de Tehnologia Informației - OTI
- Olimpiada de Inovare și Creație Digitală - Infoeducație
- Olimpiada de Informatică Aplicată - Acadnet
- Olimpiada de Securitate Cibernetică - ONS
- Olimpiada Națională de Creativitate Științifică (Juniori și Seniori Echipe)
- Concursul Pluridisciplinar „Prosoft@Nt”
- Concursul de Robotică „Nextlab.Tech”
- Concursul Interdisciplinar „Literatura Română în Dimensiune Virtuală”
- Concursul Interdisciplinar „Geoinformatica”

- Competiția „Technovation Girls”
- Concursul Interdisciplinar „Grigore C. Moisil”
- Concursul Național Interdisciplinar „Geo-Info Virtual”
- Concursul Național „comunicare.ortografie.ro”
- Concursul Național Integrat de Matematică și Informatică „Sever Aurel Groze”
- Concursul Național „Mateinfoub" Secțiunile Matematică și Informatică
- Concursul de Robotică Școlară „First Tech Challenge”
- Competiția Arduino
- Concursul Național Bebras România
- Concursul Național de Minecraft pentru Educație
- Concursul Național Robochallenge

Câteva exemple de abordări multi-, inter- și transdisciplinare care implică baze de date (SGBD), dar și securitate cibernetică, pagini web și programare vizuală:

- Crearea unei aplicații web care să ofere informații despre securitatea cibernetică pentru utilizatorii obișnuși. Această aplicație ar putea implica dezvoltarea unei baze de date pentru stocarea datelor despre diverse amenințări cibernetică și metode de protecție. Echipa multidisciplinară ar putea include experți în securitate cibernetică, dezvoltatori web și designeri de interfețe utilizator.
- Proiectul unei "Biblioteci Școlare On-line" ar putea implica multiple discipline precum informatică, limba și literatura română, istorie și matematică. Elevii ar putea colabora pentru a dezvolta o platformă web care să ofere acces la resurse educaționale, cărți electronice și alte materiale didactice, folosind o bază de date pentru a gestiona informațiile. Aspectele de securitate cibernetică ar putea fi incluse pentru a proteja datele și confidențialitatea utilizatorilor.
- Crearea unui "Joc Educațional Interactiv" ar putea implica informatică, matematică și limbi străine. Elevii ar putea să dezvolte un joc pe calculator sau pe o platformă web care să faciliteze învățarea matematicii și a unei limbi străine într-un mod interactiv și distractiv. Acest proiect ar putea implica utilizarea unei baze de date pentru a gestiona nivelurile jocului și progresul utilizatorilor, iar aspectele de securitate cibernetică ar putea fi integrate pentru a proteja datele utilizatorilor.
- Dezvoltarea unei platforme de blogging care să integreze aspecte de pagini web, baze de date și securitate cibernetică. Această platformă ar putea permite utilizatorilor să creeze și să

administreze bloguri personale, să publice conținut, să interacționeze cu alți utilizatori și să-și protejeze datele prin intermediul unor mecanisme de autentificare și autorizare.

- Crearea unei aplicații mobile pentru educație financiară care să implice elemente de programare vizuală, baze de date și securitate cibernetică. Această aplicație ar putea să ofere utilizatorilor informații și instrumente pentru gestionarea finanțelor personale, să utilizeze o bază de date pentru a stoca datele financiare și să implementeze măsuri de securitate pentru a proteja informațiile sensibile ale utilizatorilor.
- Dezvoltarea unei platforme de e-learning care să implice aspecte de dezvoltare web, programare vizuală și securitate cibernetică. Această platformă ar putea să ofere cursuri on-line pentru diferite subiecte, să permită participanților să-și urmărească progresul și să interacționeze cu alți utilizatori, să utilizeze o bază de date pentru a gestiona conținutul și să implementeze măsuri de securitate pentru a proteja datele personale și informațiile de învățare.
- Proiectul "Școala Viitorului" ar putea implica o gamă largă de discipline, cum ar fi informatica, științele sociale, matematica și arte vizuale. Elevii ar putea să lucreze în echipă pentru a dezvolta o platformă web completă care să ofere o experiență holistică de învățare, inclusiv cursuri on-line, resurse educaționale, instrumente de colaborare și comunitate virtuală. Acest proiect ar putea integra o bază de date pentru a gestiona toate aspectele platformei, iar securitatea cibernetică ar trebui să fie o preocupare principală pentru a proteja datele personale ale utilizatorilor și integritatea sistemului în sine.

Aceste abordări ar putea fi adaptate pentru a se potrivi cu obiectivele de învățare specifice, interesele și capacitățile elevilor din cadrul unei școli. Prin implicarea elevilor în proiecte care îmbină mai multe discipline și aspecte practice, aceștia pot dezvolta competențe esențiale precum colaborarea, rezolvarea de probleme și gândirea critică, în timp ce își îmbunătățesc înțelegerea asupra subiectelor studiate.

III. VALORIZAREA DE OPORTUNITĂȚI ASIGURATE PRIN PROIECTELE ȘI PROGRAMELE MINISTERULUI EDUCAȚIEI

Ministerul Educației a oferit și asigură în continuare oportunități profesorilor prin proiectele și programele implementate la nivel național. În ultimii ani, informatica și tehnologia informației și a comunicațiilor au devenit mai importante în procesul de predare-învățare-evaluare, odată cu dezvoltarea tehnologiei și a produselor de eLearning. Beneficiile se transpun în îmbunătățirea eficienței și eficacității educației la toate nivelurile, atât formale cât și informale.

Conform Strategiei Naționale privind Agenda digitală pentru România 2020 intervențiile care urmează a fi implementate în România, sub coordonarea Ministerului Educației, cu privire la educația prin informatică și tehnologia informației și a comunicațiilor, pot fi organizate în 3 categorii:

- educația prin activitate curriculară bazată pe informatică și tehnologia informației și a comunicațiilor;
- educația prin activitatea extracurriculară bazată pe informatică și tehnologia informației și a comunicațiilor;
- pregătirea profesională continuă cu ajutorul informaticii și tehnologiei informației și a comunicațiilor.

Deoarece profesorii se confruntă cu cerințe profesionale care se schimbă cu rapiditate, este necesar ca aceștia să dețină un set de competențe ce devine din ce în ce mai extins și mai sofisticat. În ultima perioadă, educația în era digitală și rolul inteligenței artificiale în procesul instructiv-educativ au căpătat noi valențe. Multe din activităților educaționale s-au extins și în mediul online. Astfel, tehnologizarea și digitalizarea procesului instructiv-educativ a dus la multe modificări în cadrul acestuia.

Uniunea Europeană și-a propus ca între 2021 și 2027 să îmbunătățească educația digitală. Cele două priorități propuse sunt:

- Încurajarea dezvoltării unui ecosistem de educație digitală de înaltă performanță;
- Dezvoltarea aptitudinilor și competențelor digitale relevante pentru transformarea digitală.

III.1. Discipline opționale, din oferta națională, care completează curriculumul obligatoriu în domeniul disciplinei/modulului de studiu, în relație directă/indirectă cu acesta

Planurile cadru sunt documente elaborate de Ministerul Educației și indică, pentru fiecare filieră, profil, specializare și clasă, disciplinele de studiu și numărul orelor/săptămână pentru fiecare dintre acestea.

Planurile cadru pentru ciclurile inferior și superior ale liceului au următoarele componente: trunchi comun, curriculum diferențiat și curriculum la decizia școlii. Trunchiul comun și curriculumul diferențiat cuprind disciplinele care se studiază în mod obligatoriu în cadrul unei specializări. Curriculumul la decizia școlii menționează oferta curriculară proprie unității de învățământ, care o completează pe cea obligatorie. Disciplinele opționale cuprinse în curriculumul la decizia școlii pot fi propuse la nivel național sau la nivelul școlii.

Tipurile de opționale care pot fi incluse în curriculumul la decizia școlii sunt:

- *Opțional de aprofundare* - are ca scop realizarea unui parcurs suplimentar pentru dezvoltarea competențelor specifice prevăzute de programa școlară a unei discipline de trunchi comun, prin noi activități de învățare.
- *Opțional ca nouă disciplină/nou domeniu de studiu* - are ca scop realizarea unor noi achiziții, specifice nevoilor și intereselor de învățare ale elevilor, diferite de cele stipulate în trunchiul comun și, după caz, în curriculumul diferențiat.

Din oferta națională de discipline opționale se distinge **„Introducere în învățarea automată (machine learning)”**, care poate fi o completare de 1 oră/săptămână pentru elevii din clasa a XII-a care studiază disciplina informatică.

În prezentarea programei se precizează că învățarea automată are ca scop principal construirea de sisteme inteligente, care reprezintă obiectul de studiu al inteligenței artificiale. Învățarea automată are la bază algoritmi care determină sistemele să învețe din propria experiență, nu doar să se bazeze pe algoritmi predefiniți. În acest context, programa școlară a disciplinei **„Introducere în învățarea automată (machine learning)”** vizează două competențe generale: implementarea algoritmilor în limbajul Python și elaborarea creativă de soluții informatice care vizează aspecte sociale, culturale și personale, utilizând algoritmi care au la bază învățarea automată.

Conținuturile sunt organizate în două părți: familiarizarea cu limbajul de programare Python și prezentarea conceptelor de bază ale învățării automate. Sunt recomandate medii de dezvoltare și

platforme care pot fi utilizate pentru implementarea algoritmilor utilizând limbajul Python.

Activitățile de învățare exemplificate în programa școlară sunt ancorate în realitate. Sunt recomandate activități de evaluare care să încurajeze creativitatea elevilor. Programa prezintă, de asemenea, o listă de resurse gratuite disponibile on-line.

III.2. Ghiduri elaborate în cadrul unor proiecte

Ghidul profesorului din România

Acest ghid a fost elaborat în cadrul proiectului necompetitiv sistemic al Ministerului Educației, POCU/904/6/25, cod SMIS 146587, Profesionalizarea carierei didactice - PROF, care a debutat în aprilie 2021 și s-a finalizat în decembrie 2023. Dezvoltarea acestui proiect a avut loc în condițiile elaborării și adoptării unei noi Legi a învățământului preuniversitar nr. 198/2023, *cu modificările și completările ulterioare*, parte a proiectului de țară România educată.

Necesitatea acestui ghid s-a impus și în condițiile în care pedagogia modernă presupune două elemente active în procesul educațional. Pe de o parte, profesorul care manageriază demersul didactic și elevul care are o responsabilitate crescută, dinamică în acest proces, accentul punându-se pe elev. Practic, profesorul nu mai este un transmițător de cunoștințe, ci un facilitator, un îndrumător în procesul didactic, coordonând, pe baza dialogului, lucrul cu elevii săi pentru atingerea obiectivelor pe care și le-a propus. Alături de cele afirmate mai sus, noul profesor trebuie să fie creativ și inovativ, să aibă o atitudine pozitivă și critică în același timp, și să-și conducă elevii spre activități cât mai diverse.

Elaborarea acestui ghid a avut în vedere documentul Ministerului Educației, numit Repere pentru proiectarea și actualizarea Curriculumului național, document de politici educaționale care și-a propus „o perspectivă unitară și actuală în proiectarea și dezvoltarea curriculumului național (CN), care să asigure coerența acestuia pe orizontală și verticală”.

Având ca fundament Legea învățământului preuniversitar nr. 198/2023, *cu modificările și completările ulterioare*, reorganizarea curriculumului în sistem hibrid sau on-line este stipulată la art. 18 alin. (2), unde se precizează că: *Pentru asigurarea exercitării dreptului fundamental la educație, în situația în care se instituie starea de urgență, starea de alertă sau starea de asediu sau ori de câte ori Ministerul Educației dispune astfel, la cererea unității de învățământ preuniversitar, cu avizul DJIP/DMBIP, activitățile de predare, învățare și evaluare se pot desfășura și în sistem on-line sau hibrid, pentru o perioadă determinată. Utilizarea la clasă a*

sistemului blended-learning sau hibrid este o practică frecventă a majorității profesorilor, fiind din ce în ce mai mult agreată de elevi.

Pe de altă parte, avansarea tehnologiei în toate aspectele vieții a creat o nouă cultură, redefinind inclusiv modul în care predăm, învățăm și evaluăm. Sistemul de învățare de tip blended-learning este din ce în ce mai prezent și mai utilizat în toate formele de educație.

Considerată o provocare pentru profesori, predarea - învățarea și evaluarea on-line au devenit o obișnuință, odată cu antrenarea digitalului în procesul educațional ca urmare a programului „România educată”. O urmare a fost, mai ales în condițiile pandemiei din anii 2020-2022, trecerea spre *învățarea combinată (blended-learning) care reprezintă o asamblare, pe linie instrumentală și procedurală, dintre învățarea tradițională și e-learning (on-line learning, învățarea on-line)* (Adascăliței, A., Cucuș, C., Vlada, M., 2018).

În contextul proiectului PROF, ghidul profesorului contribuie la realizarea obiectivului general al proiectului, care vizează asigurarea mentoratului profesional pe durata întregii cariere didactice, în sistemul de învățământ preuniversitar, prin crearea unui sistem național coerent și fiabil de formare profesională și de dezvoltare a competenței didactice.

Acest ghid a fost elaborat ca un instrument util pentru profesorii de liceu, în scopul adaptării la specificul sistemului educațional românesc a tendințelor și perspectivelor actuale din educație, de a asigura coerență în implementarea noului curriculum, plecând de la așteptările exprimate față de elevi și prin raportare la acestea, respectiv profilul absolventului de învățământ secundar superior.

Acesta oferă o viziune integratoare asupra cadrului strategic care cuprinde sistemul de educație și formare ca un întreg, într-o perspectivă holistă și incluzând învățarea pe tot parcursul vieții. Prin exemplele de bună practică privind proiectarea și evaluarea curriculumului liceal în context blended-learning, ghidul răspunde, în același timp, la întrebarea: *Cum putem face concret trecerea de la reperatele teoretice la modalitățile de realizare practice, așa încât să fie posibile ca rezultate competențele-cheie descrise prin profilul de formare european?*

Exigențele privind adaptarea competențelor profesorilor la schimbările din structurile/procesele de educație, precum și evoluțiile din planul nevoilor de educație și al curriculum-ului educațional, fac necesară formarea continuă a personalului didactic pentru actualizarea și dezvoltarea competențelor, inclusiv dobândirea de noi competențe.

Link Ghiduri: <https://www.eprof.ro/biblioteca-virtuala/mentorat-didactic/>

BIBLIOGRAFIE

1. Bocoș, M., Jucan, D., Teoria și metodologia instruirii. Teoria și metodologia evaluării: repere și instrumente didactice pentru formarea profesorilor, Ed. Paralela 45, 2019
2. Brut, M., Instrumente pentru e-learning, Ed. Polirom, 2006
3. Cerchez, E., Șerban, M., Programarea în limbajul C/C++ pentru liceu, vol I-IV, Ed. Polirom, 2004-2013
4. Cerghit, I., Metode de învățământ, Ed. Polirom, 2006
5. Cerghit, I., Sisteme de instruire alternative și complementare. Structuri, stiluri și strategii, Ed. Polirom, 2008
6. Cormen, T., Leiserson, Ch., Rivest, R. Introducere în algoritmi, Ed. Byblos, Cluj, 2004
7. Cucuș, C., Pedagogie, Ed. Polirom, 2014
8. Cucuș, C., Psihopedagogie pentru examenele de definitivare și grade didactice, Ed. Polirom, 2016
9. Cucuș, C., Teoria și metodologia evaluării, Ed. Polirom, 2008
10. Economou, A., SELFIEforTEACHERS Toolkit - Using SELFIEforTEACHERS, Publications Office of the European Union, Luxembourg, 2023, doi:10.2760/626409, JRC129699
11. Gălățan, C., Secrete C++, Ed. Microinformatica, 2006
12. Giumale, C., Negreanu, L., Călinoiu, S., Proiectarea și analiza algoritmilor. Algoritmi de sortare, Ed. ALL, 1997
13. Holotescu, C., Grossecck, G., Educație deschisă. Resurse educaționale deschise și cursuri on-line masive deschise, Polirom,
14. Ionescu, M., Bocoș, M.D. (coord.), Tratat de didactică modernă, Ed. Paralela 45, 2017
15. Jinga, I., Istrate, E., Instruirea și evaluarea asistată de calculator, Ed. ALL, 2006
16. Knuth, D.E., Arta programării calculatoarelor vol.2, Algoritmi seminumerici, Ed. Teora, 2000
17. Knuth, D.E., Arta programării calculatoarelor, vol.1, Algoritmi fundamentali, Ed. Teora, 1999
18. Knuth, D.E., Arta programării calculatoarelor, vol.3, Sortare și căutare, Ed. Teora, 2001
19. Manolescu, M., Evaluarea școlară, Ed. Meteor, 2006
20. Masalagiu, C., Asiminoae, A., Țibu, M., Didactica predării informaticii, Ed. Polirom, 2016
21. Mitrana, V., Provocarea algoritmilor, Ed. Agni, 1994
22. Nicolescu, B., Nous, la particule et le monde, Editions Le Mail, Paris, 1985
23. Oprea, C.L., Strategii didactice interactive, Ed. Didactică și pedagogică, 2006 Repere metodologice pentru aplicarea curriculumului la clasa a IX-a, anul școlar 2021-2022 Informatică
24. Petre, C., Popa, D. șa, Metodica predării informaticii și tehnologiei informației, Ed. Arves, 2002

25. Popovici Borzea, A., Integrarea curriculară și dezvoltarea capacităților cognitive, Ed. Polirom, 2017
26. Potolea, D., Neacșu, I., Iucu, R.B., Pânișoară, I.O. (coord), Pregătirea psihopedagogică. Manual pentru definitivat și gradul didactic II, Ed. Polirom, 2008
27. Stoica, A. (coord.), Evaluarea curentă și examenele, Ghid pentru profesori, Ed. Prognosis, 2001
28. Stoica, A., Evaluarea progresului școlar. De la teorie la practică. Ed. Humanitas, 2003
- *** Anexă din 5 februarie 2021 privind documentul de politici educaționale Repere pentru proiectarea, actualizarea și evaluarea Curriculumului național. Cadrul de referință al Curriculumului național, publicată în MONITORUL OFICIAL nr. 151 bis din 15 februarie 2021
- *** Cadrul european pentru competența digitală a profesorilor: DigCompEdu, Traducere și adaptare în limba română: *Fundația EOS România și Coaliția pentru Educație Digitală*
- *** Didactica formării competențelor. Cercetare - dezvoltare - inovare - formare, Universitatea de Vest „Vasile Goldiș” din Arad, Centrul de Didactică și Educație, ”Vasile Goldiș” University Press Arad, 2012
- *** Ghid de evaluare la informatică și tehnologia informației, Serviciul Național de Evaluare și Examinare, Ed. Aramis Print, 2001
- *** Ghid pentru aplicarea practicilor educaționale deschise în timpul pandemiei de coronavirus.
- *** Ordin nr. 3410 din 16 martie 2009, publicat în MONITORUL OFICIAL nr. 545/2009
- *** Ordin nr. 3667 din 13 februarie 2023 pentru modificarea Ordinului ministrului educației, cercetării și inovării nr. 3.410/2009 privind aprobarea planurilor-cadru de învățământ pentru clasele a IX-a - a XII-a, filierele teoretică și vocațională, cursuri de zi, publicat în MONITORUL OFICIAL nr. 162 din 24 februarie 2023
- *** Ordin nr. 4049 din 16 iunie 2022 privind aprobarea Programei școlare pentru disciplina opțională „Introducere în învățarea automată (machine learning)” — curriculum la decizia școlii pentru învățământul liceal, clasa a XI-a/clasa a XII-a, publicat în MONITORUL OFICIAL nr. 644 din 29 iunie 2022
- *** Ordin nr. 4150/29 iunie 2022 pentru aprobarea cadrului de competențe digitale al profesionistului din educație.
- *** Programul național de Dezvoltare a Competențelor de Evaluare ale Cadrelor Didactice (DeCeE), Centrul Național pentru Curriculum și Evaluare în Învățământul Preuniversitar - CNCEIP, Ed. Euro Standard, 2008
- *** Utilizarea Resurselor Educaționale Deschise în conformitate cu Recomandările UNESCO, Mai 2020

RESURSE WEB

https://www.edu.ro/sites/default/files/fi%C8%99iere/Legislatie/2023/OM_3800_2023_structura_an_scolar_2023_2024.pdf

http://proiecte.pmu.ro/c/document_library/get_file?p_1_id=14661&folderId=18026&name=DLFE-1201.pdf

<https://epale.ec.europa.eu/sites/default/files/strategia-nationala-agenda-digitala-pentru-romania-20202c-20-feb.2015.pdf>

<https://legislatie.just.ro/Public/DetaliiDocumentAfis/250235>

<https://eos.ro/portofoliu/cadrul-european-pentru-competenta-digitala-a-profesorilor-digcompedu/>

<https://eos.ro/wp->

[content/uploads/2022/10/eos_cadrul_european_pentru_competenta_digitala_a_profesorilor - digcompedu fin_002.pdf](content/uploads/2022/10/eos_cadrul_european_pentru_competenta_digitala_a_profesorilor_-_digcompedu_fin_002.pdf)

<https://legislatie.just.ro/Public/DetaliiDocument/257484>

<https://educators-go-digital.jrc.ec.europa.eu/>

<https://education.ec.europa.eu/selfie-for-teachers>

<https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32006H0962&from=EN>

<https://digitaledu.ro/resurse-educationale-deschise/>

COLECTIV DE AUTORI

Coordonator

Livia Demetra ȚOCA, Centrul Național de Politici și Evaluare în Educație

Monitorizare:

Grigore ALBEANU, Universitatea "Spiru Haret", București

Nușa DUMITRIU-LUPAN, Liceul „Radu Miron” Vaslui,

Clubul elevilor „Spiru Haret”, Bârlad

Alexandra FORTIȘ, Universitatea de Vest, Facultatea de Matematică și Informatică

Adrian IFTENE, Universitatea „Alexandru Ioan Cuza”, Facultatea de Informatică

Autori

Cristina Elena ANTON, Colegiul Național "Gheorghe Munteanu Murgoci" Brăila

Alina Gabriela BOCA, Colegiul Național de Informatică "Tudor Vianu" București

Alina PINTESCU, Colegiul Național "Gheorghe Șincai" Baia Mare

Florentina UNGUREANU, Colegiul Național de Informatică Piatra Neamț