Anexa nr. 13

Programe de examen
pentru disciplina

Informatică

Examenul național de bacalaureat - 2030
Învățământ liceal

– 2026 –

TABEL CENTRALIZATOR AL PROGRAMELOR DE EXAMEN

Nr. crt.	Cod programă de examen (BAC)	Proba	Raportare (TC / TC+CS / CS)	Filiera	Profilul	Specializarea/Domeniul/ Calificările profesionale
1.	INF_P1	E – a doua probă	CS	Teoretică	Real	Specializarea matematică-informatică
2.	INF_P2	E – a doua probă	CS	Teoretică	Real	Specializarea științe ale naturii
3.	INF_P3	E – a doua probă	CS	Vocațională	Militar	Specializarea matematică-informatică militară

Anexa nr. 13.1. Programa de examen pentru disciplina *Informatică* – INF_P1 (CS)

- *Proba E – a doua probă (probă scrisă, specifică profilului/specializării)*
- *Filiera teoretică, profilul real, specializarea matematică-informatică*

PREAMBUL

Programa de examen pentru disciplina *Informatică* fundamentează elaborarea variantelor de subiecte și a baremelor de evaluare și de notare pentru **proba E – a doua probă** din cadrul examenului național de bacalaureat 2030. Prezenta programă precizează **competențele de evaluat**, **domeniile de conținut** și **conținuturile** vizate de evaluarea specifică acestui examen pentru candidații de la specializarea matematică-informatică (incluzând clase de la această specializare cu predarea disciplinei informatică în regim intensiv).

Programa de examen derivă din **programele școlare în vigoare pentru învățământul liceal**, aprobate prin **Ordinul ministrului educației și cercetării nr. 6930/2025 și Ordinul ministrului educației și cercetării nr. 4370/2026**. Evaluarea vizează competențele specifice care contribuie la formarea/dezvoltarea competențelor generale pe parcursul întregului ciclu liceal, prin raportare la profilul de formare al absolventului.

Obiectivul evaluării prin proba de examen la această disciplină este de a constata măsura în care candidatul *recunoaște, explică, utilizează, analizează, evaluează și creează* în domeniul informatic, în situații de complexitate corespunzătoare raportării (*CS*) și tipului de probă.

Prin susținerea probei de examen la disciplina informatică, candidatul este pus în situația de a face dovada următoarelor **competențe generale (CG)**, dobândite pe parcursul învățământului liceal (clasele a IX-a – a XII-a):

CG1	Identifică principalele caracteristici ale modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru înțelegerea fundamentelor programării
CG2	Explică principii care stau la baza modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru a fundamenta în mod logic proiectarea și implementarea soluțiilor informatice
CG3	Utilizează modele conceptuale și operaționale ale dezvoltării produselor software, în scopul obținerii de soluții informatice funcționale și eficiente
CG4	Analizează caracteristicile și aplicabilitatea modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru a selecta soluțiile cele mai potrivite în funcție de contextul informatic dat
CG5	Evaluează corectitudinea și eficiența soluțiilor informatice, în vederea optimizării și asigurării funcționalității în diverse scenarii de utilizare
CG6	Elaborează algoritmi și programe personalizate, pentru a crea soluții informatice coerente și adaptate cerințelor

Prin variantele de subiecte elaborate pentru examenul național de bacalaureat se evaluează competențele și conținuturile din prezenta programă, iar baremele de evaluare și de notare prevăd acordarea punctajelor pentru **orice modalitate corectă de rezolvare** a cerințelor.

COMPETENȚE DE EVALUAT

Fiecărei competențe generale îi sunt asociate **competențele specifice (CS)** care pot fi evaluate prin examen, selectate din programele școlare în vigoare și prezentate cumulativ pentru ciclul liceal:

CG1 – Identifică principalele caracteristici ale modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru înțelegerea fundamentelor programării

Cod CS	Competențe specifice de evaluat
IX.CS1.1.	Identifică principalele caracteristici ale organizării datelor în cadrul unor modele conceptuale fundamentale - date simple sau liste, pentru a structura și accesa date în vederea prelucrării acestora
X.CS1.1.	Identifică principalele caracteristici ale organizării datelor în cadrul unor modele conceptuale simple - neliniare, liniare, asociative, mixte, pentru a structura și accesa date în vederea prelucrării acestora

Cod CS	Competențe specifice de evaluat
XI.CS1.1.	Identifică principalele caracteristici ale organizării datelor în cadrul unor modele conceptuale complexe – rețea, ierarhice, obiectuale, pentru a structura și accesa date în vederea prelucrării acestora
IX.CS1.2.	Identifică specificul, caracteristicile și etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea algoritmică a numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS1.2.	Identifică specificul, caracteristicile și etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea listelor
XI.CS1.2.	Identifică specificul, caracteristicile și etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea grafurilor, arborilor
IX.CS1.3.	Identifică specificul, caracteristicile și etapele unor strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS1.3.	Identifică specificul, caracteristicile și etapele unor strategii de tip Divide et impera, Greedy pentru rezolvarea problemelor
XI.CS1.3.	Identifică specificul, caracteristicile și etapele unor strategii de tip Backtracking pentru rezolvarea problemelor
IX.CS1.4.	Identifică principalele elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea rezolvării eficiente a problemelor informatice
X.CS1.4.	Identifică principalele elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele simple - neliniare, liniare, asociative, mixte, în vederea rezolvării eficiente a problemelor informatice
XI.CS1.4.	Identifică principalele elemente ale limbajului de programare utilizate pentru memorarea și prelucrarea datelor organizate în modele complexe - rețea, ierarhice, în vederea rezolvării eficiente a problemelor informatice
IX.CS1.5.	Identifică elementele de sintaxă și componentele din definiția și apelul subprogramelor și subprogramelor predefinite pentru operații uzuale, pentru a le utiliza în implementarea algoritmilor în limbaj de programare
X.CS1.5.	Identifică elementele de sintaxă și componentele din definiția și apelul subprogramelor recursive, pentru a le utiliza în implementarea algoritmilor în limbaj de programare
XI.CS1.5.	Identifică elementele de sintaxă și componentele din definiția claselor, pentru a le utiliza în implementarea algoritmilor în limbaj de programare cu programare orientată pe obiecte
XII.CS1.1.	Identifică principalele caracteristici ale organizării datelor în cadrul unor modele conceptuale avansate - pentru proiectarea bazelor de date sau învățare automată, pentru a structura și accesa date în vederea prelucrării acestora
XII.CS1.2.	Identifică specificul, caracteristicile și etapele unor strategii de normalizare a modelului conceptual al unei probleme de gestiune
XII.CS1.4.	Identifică principalele comenzi SQL și ale limbajului de programare pentru a identifica soluția adecvată prelucrării datelor organizate în baze de date, în vederea rezolvării eficiente a problemelor informatice de gestiune
XII.CS1.3.	Identifică specificul, caracteristicile și etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în învățarea automată
XII.CS1.5.	Identifică principalele elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în învățarea automată, în vederea rezolvării eficiente a problemelor informatice

CG2 – Explică principii care stau la baza modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru a fundamenta în mod logic proiectarea și implementarea soluțiilor informatice

Cod CS	Competențe specifice de evaluat
IX.CS2.1.	Explică principiile de organizare a datelor în cadrul unor modele conceptuale fundamentale - date simple sau liste, pentru a le gestiona eficient
X.CS2.1.	Explică principiile de organizare și prelucrare a datelor în cadrul unor modele conceptuale simple - neliniare, liniare, asociative, mixte, pentru a le gestiona eficient
XI.CS2.1.	Explică principiile de organizare a datelor în cadrul unor modele conceptuale complexe - rețea, ierarhice, obiectuale, pentru a le gestiona eficient
IX.CS2.2.	Explică etapele algoritmilor specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea algoritmică a numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS2.2.	Explică etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea listelor
XI.CS2.2.	Explică etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea grafurilor, arborilor
IX.CS2.3.	Explică etapele unor strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS2.3.	Explică etapele unor strategii de tip Divide et impera, Greedy pentru rezolvarea problemelor
XI.CS2.3.	Explică etapele unor strategii de tip Backtracking pentru rezolvarea problemelor
IX.CS2.4.	Explică rolul principalelor elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program în vederea rezolvării eficiente a problemelor informatice
X.CS2.4.	Explică rolul principalelor elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele simple - neliniare, liniare, asociative, mixte, în vederea rezolvării eficiente a problemelor informatice
XI.CS2.4.	Explică rolul principalelor elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele complexe - rețea, ierarhice, în vederea rezolvării eficiente a problemelor informatice
IX.CS2.5.	Explică mecanismul de executare a subprogramelor și subprogramelor predefinite pentru operații uzuale, pentru a le utiliza în implementarea algoritmilor în limbaj de programare
X.CS2.5.	Explică mecanismul de executare a subprogramelor recursive, pentru a le utiliza în implementarea algoritmilor în limbaj de programare
XI.CS2.5.	Explică rolul restricțiilor de acces și al componentelor unei clase, pentru a le utiliza în proiectarea claselor și implementarea algoritmilor cu programare orientată pe obiecte
XII.CS2.1.	Explică principiile de organizare a datelor în cadrul unor modele conceptuale avansate - pentru proiectarea bazelor de date sau învățare automată, pentru a le gestiona eficient
XII.CS2.2.	Explică etapele unor strategii de normalizare a modelului conceptual al unei probleme de gestiune
XII.CS2.4.	Explică rolul componentelor principalelor comenzi SQL și ale limbajului de programare pentru a identifica soluția adecvată prelucrării datelor organizate în baze de date, în vederea rezolvării eficiente a problemelor informatice de gestiune
XII.CS2.3.	Explică etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în învățarea automată
XII.CS2.5.	Explică rolul principalelor elemente ale limbajului de programare utilizate pentru prelucrarea datelor în învățarea automată, în vederea rezolvării eficiente a problemelor informatice

CG3 – Utilizează modele conceptuale și operaționale ale dezvoltării produselor software, în scopul obținerii de soluții informatice funcționale și eficiente

Cod CS	Competențe specifice de evaluat
IX.CS3.1.	Utilizează modul de organizare și prelucrare a datelor în cadrul unor modele conceptuale fundamentale - date simple sau liste, pentru a rezolva probleme de gestionare a datelor

Cod CS	Competențe specifice de evaluat
X.CS3.1.	Utilizează modul de organizare și prelucrare a datelor în cadrul unor modele conceptuale simple - neliniare, liniare, asociative, mixte, pentru a rezolva probleme de gestionare a datelor
XI.CS3.1.	Utilizează modul de organizare și prelucrare a datelor în cadrul unor modele conceptuale complexe - rețea, ierarhice, obiectuale, pentru a rezolva probleme de gestionare a datelor
IX.CS3.2.	Aplică algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea algoritmică a numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS3.2.	Aplică algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea listelor
XI.CS3.2.	Aplică algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea grafurilor, arborilor
IX.CS3.3.	Aplică strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS3.3.	Aplică strategii de tip Divide et impera, Greedy pentru rezolvarea problemelor
XI.CS3.3.	Aplică strategii de tip Backtracking pentru rezolvarea problemelor
IX.CS3.4.	Utilizează principalele elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea rezolvării eficiente a problemelor informatice
X.CS3.4.	Utilizează principalele elemente ale limbajului de programare pentru prelucrarea datelor organizate în modele simple - neliniare, liniare, asociative, mixte, în vederea rezolvării eficiente a problemelor informatice
XI.CS3.4.	Utilizează principalele elemente ale limbajului de programare pentru prelucrarea datelor organizate în modele complexe - rețea, ierarhice, în vederea rezolvării eficiente a problemelor informatice
IX.CS3.5.	Utilizează elementele de sintaxă și componentele din definiția și apelul subprogramelor și subprogramelor predefinite pentru operații uzuale, în implementarea algoritmilor în limbaj de programare
X.CS3.5.	Utilizează elementele de sintaxă și componentele din definiția și apelul subprogramelor recursive, în implementarea algoritmilor în limbaj de programare
XI.CS3.5.	Utilizează elementele de sintaxă și componentele din definiția claselor, în implementarea algoritmilor în limbaj de programare cu programare orientată pe obiecte
XII.CS3.1.	Utilizează modul de organizare și prelucrare a datelor în cadrul unor modele conceptuale avansate – la proiectarea bazelor de date sau la învățare automată, pentru a rezolva probleme de gestionare a datelor
XII.CS3.2.	Aplică strategii de normalizare a modelului conceptual al unei probleme de gestiune
XII.CS3.4.	Utilizează principalele comenzi SQL și ale limbajului de programare pentru a identifica soluția adecvată prelucrării datelor organizate în baze de date, în vederea rezolvării eficiente a problemelor informatice de gestiune
XII.CS3.3.	Aplică algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în învățarea automată
XII.CS3.5.	Utilizează principalele elemente ale limbajului de programare pentru prelucrarea datelor în învățarea automată, în vederea rezolvării eficiente a problemelor informatice

CG4 – Analizează caracteristicile și aplicabilitatea modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru a selecta soluțiile cele mai potrivite în funcție de contextul informatic dat

Cod CS	Competențe specifice de evaluat
IX.CS4.1.	Analizează caracteristicile, modul de prelucrare, avantajele și dezavantajele utilizării unor modele conceptuale fundamentale - date simple sau liste, pentru a alege structura adecvată în rezolvarea unor probleme concrete
X.CS4.1.	Analizează caracteristicile, modul de prelucrare, avantajele și dezavantajele utilizării unor modele conceptuale simple - neliniare, liniare, asociative, mixte, pentru a alege structura adecvată în rezolvarea unor probleme concrete

Cod CS	Competențe specifice de evaluat
XI.CS4.1.	Analizează caracteristicile, modul de prelucrare, avantajele și dezavantajele utilizării unor modele conceptuale complexe - rețea, ierarhice, obiectuale, pentru a alege structura adecvată în rezolvarea unor probleme concrete
IX.CS4.2.	Analizează comportamentul, aplicabilitatea, avantajele și dezavantajele utilizării unor algoritmi specializați pe clase de probleme pentru prelucrarea numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS4.2.	Analizează comportamentul, aplicabilitatea, avantajele și dezavantajele utilizării unor algoritmi specializați pe clase de probleme pentru prelucrarea listelor
XI.CS4.2.	Analizează comportamentul, aplicabilitatea, avantajele și dezavantajele utilizării unor algoritmi specializați pe clase de probleme pentru prelucrarea grafurilor, arborilor
IX.CS4.3.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS4.3.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor strategii de tip Divide et impera, Greedy pentru rezolvarea problemelor
XI.CS4.3.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor strategii de tip Backtracking pentru rezolvarea problemelor
IX.CS4.4.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor elemente ale limbajului de programare, pentru a identifica soluția adecvată prelucrării datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea rezolvării eficiente a problemelor informatice
X.CS4.4.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor elemente ale limbajului de programare, pentru a identifica soluția adecvată prelucrării datelor organizate în modele simple - neliniare, liniare, asociative, mixte, în vederea rezolvării eficiente a problemelor informatice
XI.CS4.4.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor elemente ale limbajului de programare, pentru a identifica soluția adecvată prelucrării datelor organizate în modele complexe - rețea, ierarhice, în vederea rezolvării eficiente a problemelor informatice
IX.CS4.5.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării subprogramelor, în implementarea algoritmilor în limbaj de programare
X.CS4.5.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării subprogramelor recursive, în implementarea algoritmilor în limbaj de programare
XI.CS4.5.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării programării orientate pe obiecte, în implementarea algoritmilor în limbaj de programare
XII.CS4.1.	Analizează caracteristicile, modul de prelucrare, avantajele și dezavantajele utilizării unor modele conceptuale avansate – pentru proiectarea bazelor de date sau pentru învățare automată, pentru a alege structura adecvată în rezolvarea unor probleme concrete
XII.CS4.2.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor strategii de normalizare a modelului conceptual al unei probleme de gestiune
XII.CS4.4.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării comenzilor SQL și a elementelor specifice limbajului de programare, în vederea rezolvării eficiente a problemelor informatice de gestiune
XII.CS4.3.	Analizează comportamentul, aplicabilitatea, avantajele și dezavantajele utilizării unor algoritmi specializați pe clase de probleme pentru învățarea automată
XII.CS4.5.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor elemente ale limbajului de programare, pentru a identifica soluția adecvată prelucrării datelor în învățarea automată, în vederea rezolvării eficiente a problemelor informatice

CG5 – Evaluează corectitudinea și eficiența soluțiilor informatice, în vederea optimizării și asigurării funcționalității în diverse scenarii de utilizare

Cod CS	Competențe specifice de evaluat
IX.CS5.1.	Argumentează alegerea unor modele conceptuale fundamentale - date simple sau liste - și a modurilor de prelucrare a acestora pentru rezolvarea unor probleme informatice concrete
X.CS5.1.	Argumentează alegerea unor modele conceptuale simple - neliniare, liniare, asociative, mixte și a modurilor de prelucrare a acestora pentru rezolvarea unor probleme informatice concrete
XI.CS5.1.	Argumentează alegerea unor modele conceptuale complexe - rețea, ierarhice, obiectuale și a modurilor de prelucrare a acestora pentru rezolvarea unor probleme informatice concrete
IX.CS5.2.	Evaluează corectitudinea și eficiența soluțiilor cu algoritmi specializați pe clase de probleme pentru prelucrarea numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS5.2.	Evaluează corectitudinea și eficiența soluțiilor cu algoritmi specializați pe clase de probleme pentru prelucrarea listelor
XI.CS5.2.	Evaluează corectitudinea și eficiența soluțiilor cu algoritmi specializați pe clase de probleme pentru prelucrarea grafurilor, arborilor
IX.CS5.3.	Evaluează corectitudinea și eficiența algoritmilor care utilizează strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS5.3.	Evaluează corectitudinea și eficiența algoritmilor care utilizează strategii de tip Divide et impera, Greedy pentru rezolvarea problemelor
XI.CS5.3.	Evaluează corectitudinea și eficiența algoritmilor care utilizează strategii de tip Backtracking pentru rezolvarea problemelor
IX.CS5.4.	Evaluează corectitudinea și eficiența aplicațiilor care utilizează elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea rezolvării eficiente a problemelor informatice
X.CS5.4.	Evaluează corectitudinea și eficiența aplicațiilor care utilizează elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele simple - neliniare, liniare, asociative, mixte, în vederea rezolvării eficiente a problemelor informatice
XI.CS5.4.	Evaluează corectitudinea și eficiența aplicațiilor care utilizează elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele complexe - rețea, ierarhice, în vederea rezolvării eficiente a problemelor informatice
IX.CS5.5.	Evaluează corectitudinea și eficiența programelor care utilizează subprograme, pentru a rezolva probleme informatice
X.CS5.5.	Evaluează corectitudinea și eficiența programelor care utilizează subprograme recursive, pentru a rezolva probleme informatice
XI.CS5.5.	Evaluează corectitudinea și eficiența programelor care utilizează programare orientată pe obiecte, pentru a rezolva probleme informatice
XII.CS5.1.	Argumentează alegerile făcute în cadrul proiectării unor modele conceptuale avansate - pentru proiectarea bazelor de date sau pentru învățare automată, în vederea rezolvării unor probleme informatice concrete
XII.CS5.2.	Evaluează corectitudinea și eficiența modelelor conceptuale ale unei probleme de gestiune din punctul de vedere al normalizării
XII.CS5.4.	Evaluează corectitudinea și eficiența secvențelor de cod sau aplicațiilor care utilizează principalele comenzi SQL și ale limbajului de programare pentru prelucrarea datelor organizate în baze de date, în vederea rezolvării eficiente a problemelor informatice de gestiune
XII.CS5.3.	Evaluează corectitudinea și eficiența soluțiilor cu algoritmi specializați pe clase de probleme pentru învățare automată

Cod CS	Competențe specifice de evaluat
XII.CS5.5.	Evaluează corectitudinea și eficiența aplicațiilor care utilizează elemente specifice de limbaj de programare pentru prelucrarea datelor în învățarea automată, în vederea rezolvării eficiente a problemelor informatice

CG6 – Elaborează algoritmi și programe personalizate, pentru a crea soluții informatice coerente și adaptate cerințelor

Cod CS	Competențe specifice de evaluat
IX.CS6.1.	Proiectează algoritmi personalizați care utilizează modele conceptuale fundamentale - date simple sau liste pentru organizarea și prelucrarea eficientă a datelor, utilizând algoritmi de bază, în vederea rezolvării unor probleme
X.CS6.1.	Proiectează algoritmi personalizați care utilizează modele conceptuale simple - neliniare, liniare, asociative, mixte, pentru organizarea și prelucrarea eficientă a datelor, utilizând algoritmi de bază, în vederea rezolvării unor probleme
XI.CS6.1.	Proiectează algoritmi personalizați care utilizează modele conceptuale complexe - rețea, ierarhice, obiectuale, pentru organizarea și prelucrarea eficientă a datelor, utilizând algoritmi de bază, în vederea rezolvării unor probleme
IX.CS6.2.	Proiectează soluții cu aplicarea personalizată a algoritmilor specializați pe clase de probleme pentru prelucrarea numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS6.2.	Proiectează soluții cu aplicarea personalizată a algoritmilor specializați pe clase de probleme pentru prelucrarea listelor
IX.CS6.3.	Proiectează modular algoritmi personalizați pentru rezolvarea problemelor
X.CS6.3.	Proiectează algoritmi cu aplicarea personalizată a strategiilor de tip Divide et impera, Greedy pentru rezolvarea problemelor
XI.CS6.3.	Proiectează algoritmi cu aplicarea personalizată a strategiilor de tip Backtracking pentru rezolvarea problemelor
IX.CS6.4.	Implementează aplicații care integrează în mod personalizat elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea realizării unor soluții funcționale și performante
X.CS6.4.	Implementează aplicații care integrează în mod personalizat elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele simple - neliniare, liniare, asociative, mixte, în vederea realizării unor soluții funcționale și performante
XI.CS6.4.	Implementează aplicații care integrează în mod personalizat elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele complexe - rețea, ierarhice, în vederea realizării unor soluții funcționale și performante
IX.CS6.5.	Implementează aplicații în limbaj de programare care integrează subprograme personalizate, pentru a modulariza și organiza eficient codul în cadrul soluțiilor informatice
X.CS6.5.	Implementează aplicații în limbaj de programare care integrează subprograme recursive personalizate, pentru a modulariza și organiza eficient codul în cadrul soluțiilor informatice

DOMENII DE CONȚINUT ȘI CONȚINUTURI

Conținuturile evaluate sunt organizate pe **domenii de conținut** și sunt prezentate cumulativ pentru ciclul liceal, în acord cu programele școlare în vigoare:

Domenii de conținut	Conținuturi
1. Organizarea conceptuală a datelor	**1.1. Modelul conceptual liniar - listă** - caracteristici, din punctul de vedere conceptual, ale unei liste și ale cazurilor particulare date de tipul de acces (stivă, coadă, respectiv cu acces direct, cu acces secvențial), rolul avut în prelucrarea datelor (listă de frecvențe); - repere pentru parcurgerea elementelor și pentru aplicarea algoritmilor de bază pentru prelucrarea datelor organizate liniar, cu sau fără memorare.

Domenii de conținut	Conținuturi
	1.2. Modelul conceptual neliniar - mulțime - caracteristici și repere pentru acces la date, parcurgerea lor și pentru aplicarea algoritmilor de bază în scopul prelucrării acestora. **1.3. Modelul conceptual liniar – șir de caractere** - caracteristici și repere pentru acces la date, parcurgerea lor și pentru aplicarea algoritmilor de bază în scopul prelucrării acestora. **1.4. Modelul conceptual asociativ – dicționar** - caracteristici și repere pentru acces la date, parcurgerea lor și pentru aplicarea algoritmilor de bază în scopul prelucrării acestora. **1.5. Modelul conceptual mixt (de exemplu, liste de liste, liste de șiruri de caractere)** - caracteristici și repere pentru acces la date, parcurgerea lor și pentru aplicarea algoritmilor de bază sau specifici în scopul prelucrării acestora (de exemplu, actualizarea datelor, modificarea numărului de componente ale colecției, construirea colecției). **1.6. Modelul conceptual rețea - graf** - concepte de bază și caracteristici: graf neorientat și orientat, nod și vârf, muchie și arc, adiacență, incidență, grad, grad interior, grad exterior, lanț și drum, lanț elementar și drum elementar, lanț simplu și drum simplu, lanț compus și drum compus, ciclu și circuit, ciclu elementar și circuit elementar; - metode de reprezentare a unui graf: diagramă, matrice de adiacență, liste de adiacență, liste de succesori, liste de predecesori, liste de muchii, liste de arce; - concepte de bază și caracteristici ale unor grafuri asociate unui graf: subgraf, graf parțial; - concepte de bază, caracteristici și proprietăți ale unor tipuri de grafuri: graf complet, graf conex, graf tare conex (matricea drumurilor), graf ponderat (matricea costurilor), graf hamiltonian (lanț hamiltonian, ciclu hamiltonian), graf eulerian (lanț eulerian, ciclu eulerian); - repere pentru aplicarea algoritmilor de bază în prelucrarea grafurilor. **1.7. Model conceptual ierarhic - arbore** - caracteristici și definiții echivalente ale unui arbore, respectiv arbore cu rădăcină; - concepte de bază într-un arbore cu rădăcină și caracteristici ale acestora: rădăcină, nivel al unui nod, nod frunză, nod descendent direct/fiu, nod descendent, nod ascendent direct/tată, nod ascendent, nod frate; - metode de reprezentare a unui arbore cu rădăcină: diagramă, referințe ascendente, referințe descendente; - caracteristici ale arborilor ordonați, arborilor binari (cu subtipul arbore binar de căutare) - metode de reprezentare a unui arbore binar cu referințe descendente (liste de fii); - repere pentru aplicarea algoritmilor de bază pentru prelucrarea arborilor. **1.8. Model conceptual obiectual – clase și obiecte** - concepte avansate și caracteristici: clasă, obiect, moștenire, clasă de bază, clasă derivată; - repere pentru aplicarea algoritmilor de bază cu date organizate obiectual. **1.9. Modelul conceptual relațional – entitate-relație** - concepte de bază și caracteristici ale modelului conceptual entitate-relație: entitate (cu atribute opționale, obligatorii, volatile, respectiv cu identificatori unici, tipuri de identificatori unici – simplu/compus, natural/artificial), instanță, relație (opționalitate, cardinalitate, tipuri de relații din punctul de vedere al cardinalității, relații barate, relații ierarhice, relații recursive); - repere pentru identificarea entităților și reprezentarea convențională a diagramei entitate-relație (ERD), rezolvarea relațiilor M:M (mai mulți la mai mulți), reprezentarea istoricului datelor; - concepte de bază și caracteristici ale modelului fizic: tabelă (cu câmpuri, tipuri de date, cheie primară, chei externe, înregistrări, constrângeri, reguli de integritate - de entitate și referențială); - repere pentru transformarea modelului conceptual în model fizic (mapare).

Domenii de conținut	Conținuturi
	1.10. Modele conceptuale de organizare a datelor folosite în învățarea automată - concepte de bază și caracteristici ale învățării automate: date, atribute, etichete, model, antrenare (învățare), testare, predicție; - caracteristici ale seturilor de date utilizate în învățarea automată: tipuri de date, date categorice, date complete/lipsă, dimensiune și calitate a datelor; - repere pentru transformarea datelor în formate utilizabile de către algoritmii de inteligență artificială (normalizare, scalare, augmentare, codificare a datelor, tratare a datelor lipsă, organizare a datelor în fișiere sau tabele pentru prelucrare automată).
2. Strategii de rezolvare a problemelor	**2.1. Principii de elaborare a unui program** - caracteristici ale gândirii computaționale și ale principalelor etape ale elaborării unui program (analiză, proiectare, implementare, testare și depanare), repere pentru aplicarea acestor etape în rezolvarea problemelor; - caracteristici de bază, avantaje și limite ale principalelor moduri de reprezentare a algoritmilor: blocuri grafice, pseudocod, limbaj de programare (cu interpretor sau compilator); - repere pentru proiectarea modulară a algoritmilor; - criterii de elaborare a testelor (pentru validarea datelor și pentru verificarea comportamentului algoritmului), repere pentru urmărirea evoluției valorilor variabilelor în scopul identificării și tratării erorilor; - eficiența algoritmilor: caracteristici ale eficienței din punctul de vedere al spațiului de memorie și din punctul de vedere al timpului de executare, criterii pentru determinarea ordinului de complexitate a unui algoritm polinomial (notația O); - caracteristici ale unor modalități de bază de comunicare cu programul: interfață de tip consolă, fișiere. **2.2. Prelucrări ale datelor numerice** - operații specifice cu cifrele unui număr (acces la o cifră a unui număr, adăugare a unei cifre la stânga unui număr, adăugare a unei cifre la dreapta unui număr), determinarea unui divizor/multiplu al unui număr; - repere pentru prelucrarea numerelor (de exemplu parcurgerea cifrelor unui număr, a divizorilor unui număr, descompunere a unui număr în factori primi) și aplicarea algoritmilor de bază; - algoritmi elementari pentru determinarea celui mai mare divizor comun (algoritmul lui Euclid cu scăderi repetate, respectiv cu împărțiri repetate); - repere pentru transformarea unui număr dintr-o bază de numerație în altă bază de numerație; **2.3. Metode de generare sistematică a elementelor unei liste** - repere pentru generarea unor secvențe de valori (de exemplu secvențe cu proprietăți date, termeni ai unor expresii matematice, termeni ai unor șiruri recurente) și aplicarea algoritmilor de bază. **2.4. Metode de sortare a elementelor unei liste** - caracteristici ale metodei de sortare prin selecția minimului și repere pentru aplicarea metodei; - caracteristici ale metodei de sortare cu listă de frecvențe și repere pentru aplicarea metodei; - caracteristici ale metodei bulelor și repere pentru aplicarea metodei. **2.5. Metode de prelucrare a listelor sortate** - caracteristici ale metodei căutării binare și repere pentru aplicarea acesteia; - caracteristici ale metodei interclasării și repere pentru aplicarea acesteia. **2.6. Algoritmi specifici pentru prelucrarea grafurilor** - caracteristici, etape ale metodelor de parcurgere în lățime (Breadth First Search - BFS), respectiv în adâncime (Depth First Search – DFS) și repere pentru aplicarea acestora; - repere pentru utilizarea metodelor de parcurgere pentru verificarea proprietății de conexitate a unui graf neorientat și pentru determinarea componentelor conexe; - repere pentru utilizarea metodelor de parcurgere pentru determinarea matricei drumurilor/lanțurilor unui graf;

Domenii de conținut	Conținuturi
	- repere pentru utilizarea matricei drumurilor pentru verificarea proprietății de tare conexitate și determinarea componentelor tare conexe ale unui graf orientat; - caracteristici, etape ale algoritmului lui Dijkstra pentru determinarea drumurilor de cost minim într-un graf ponderat și repere pentru aplicarea acestuia; - repere pentru adaptarea metodei de parcurgere în adâncime pentru determinarea unui ciclu eulerian; - caracteristici, etape ale algoritmului lui Prim pentru determinarea arborelui de acoperire de cost minim într-un graf ponderat și repere pentru aplicarea acestuia. **2.7. Algoritmi specifici pentru prelucrarea arborilor** - caracteristici, etape ale metodelor de parcurgere a nodurilor unui arbore binar în preordine, inordine și postordine și repere pentru aplicarea acestora; - caracteristici, etape ale metodelor de căutare și de inserare a unei chei într-un arbore binar de căutare și repere pentru aplicarea acestora. **2.8. Metoda Divide et impera** - caracteristici ale metodei Divide et impera; - repere pentru aplicarea metodei. **2.9. Metoda Greedy** - caracteristici ale metodei Greedy; - repere pentru aplicarea metodei. **2.10. Metoda Backtracking** - caracteristici ale metodei Backtracking; - repere pentru aplicarea metodei de generare sistematică a unor secvențe de valori. **2.11. Forme normale pentru proiectarea modelului conceptual al unei probleme de gestiune** - caracteristici și avantaje ale primelor trei forme normale (FN1, FN2, FN3) și repere pentru utilizarea acestora în proiectarea modelului conceptual al unei probleme de gestiune. **2.12. Algoritmi utilizați în învățare automată** - concepte de bază și caracteristici: învățare nesupervizată, clusterizare; - caracteristici ale algoritmului K-means, pentru învățarea nesupervizată; - concepte de bază și caracteristici: învățare supervizată, regresie, clasificare; - caracteristici ale algoritmului de regresie liniară, pentru învățare supervizată; - caracteristici ale algoritmului KNN (k Nearest Neighbours), pentru învățare supervizată.
3. Memorarea datelor și organizarea codului în limbaj de programare	**3.1. Subprograme** - caracteristici, rol; - concepte de bază și caracteristici: antet, corp, variabile locale, variabile globale, transmitere prin intermediul parametrilor, returnare a rezultatelor, apel, mecanism de executare; - sintaxă pentru definiția și apelul unui subprogram în Python; - caracteristici și repere de utilizare a unor subprograme predefinite în Python pentru operații matematice uzuale (de exemplu radical, valoare absolută, parte întreagă, rotunjire) și conversii; - caracteristici și repere de utilizare a unor subprograme predefinite în Python pentru colecții de valori (determinarea numărului de elemente, a minimului, maximului, sumei unor valori). **3.2. Subprograme recursive** - caracteristici, rol, mecanism de executare; - repere pentru scrierea subprogramelor recursive în Python. **3.3. Programare orientată pe obiecte** - noțiuni de bază necesare utilizării unor clase predefinite: clasă, membri ai clasei (date și metode), obiecte, biblioteci; - instanțiere a unei clase predefinite, acces la membrii unui obiect. - sintaxa definirii unei clase, niveluri de acces la membrii clasei, constructor, destructor/finalizer, clasă derivată în Python; - repere pentru definirea unor clase simple și a claselor derivate.

Domenii de conținut	Conținuturi
	3.4. Fișiere text - caracteristici, principii de lucru (deschidere, închidere, transfer de date); - funcție pentru deschiderea unui fișier în Python, clasa predefinită TextIOWrapper din Python pentru prelucrarea fișierelor: caracteristici, metode de bază (pentru citire, scriere, închidere); - repere pentru identificarea și utilizarea altor clase și metode, adecvate pentru prelucrarea fișierelor și pregătirea datelor citite din fișiere. **3.5. Clasa list din Python – clasă predefinită pentru memorarea unei liste** - caracteristici; - operatori specifici pentru acces la un element, apartenență, non-apartenență, concatenare, multiplicare, relaționare; - metode ale clasei pentru operații de bază: determinare a primei poziții a unei valori, numărare a aparițiilor unei valori, ștergere a elementului de la o anumită poziție, inserare a unei valori într-o anumită poziție, adăugare a unui element, copiere, sortare; - repere pentru identificarea și utilizarea altor metode ale clasei, adecvate pentru prelucrarea listelor. **3.6. Clasa set din Python – clasă predefinită pentru mulțimi** - caracteristici; - operații și operatori specifici: reuniune, intersecție, diferență, egalitate, incluziune, apartenență; - metode uzuale pentru adăugare, ștergere, inițializare; - repere pentru identificarea și utilizarea altor metode adecvate pentru prelucrarea mulțimilor, în funcție de nevoile proprii. **3.7. Clasa str din Python – clasă predefinită pentru șiruri de caractere** - caracteristici; - operații și operatori specifici: apartenență, concatenare, multiplicare, acces la un caracter sau subșir/secvență de caractere, comparare lexicografică; - metode uzuale pentru căutare a unei secvențe de caractere, generare a unui șir prin înlocuire a unei secvențe de caractere, separare a unor secvențe de caractere dintr-un șir; - repere pentru identificarea și utilizarea altor metode și funcții adecvate pentru prelucrarea șirurilor de caractere, în funcție de nevoile proprii. **3.8. Clasa dict din Python – clasă predefinită pentru dicționare** - caracteristici; - operații și operatori specifici: acces la date; - repere pentru identificarea și utilizarea altor metode sau operatori adecvați pentru prelucrarea dicționarelor, în funcție de nevoile proprii. **3.9. Elemente de limbaj pentru operarea cu date organizate în modele mixte** - caracteristici, acces la un element în Python. **3.10. Comenzi SQL pentru gestionarea unei baze de date** - caracteristici și sintaxă ale unor comenzi de bază pentru interogarea bazei de date (SQL): selecție, proiecție, ordonare, operatori, funcții scalare uzuale (de prelucrare a numerelor, a datelor calendaristice, a șirurilor de caractere, conversii), funcții agregate uzuale (pentru sumă, medie, numărare, minim, maxim, de tratare a valorii NULL), preluare a datelor din mai multe tabele (join); - caracteristici și sintaxă ale unor comenzi de bază pentru manipularea datelor (DML): inserare, modificare, ștergere; - caracteristici și sintaxă ale unor comenzi de bază pentru manipularea structurii bazei de date (DDL): creare a unei tabele, modificare a structurii unei tabele (adăugare a unui câmp, ștergere a unui câmp, modificare a unui câmp), ștergere a unei tabele; - repere pentru executarea comenzilor; - repere pentru identificarea și utilizarea altor comenzi, adecvate pentru prelucrarea bazelor de date, în funcție de nevoile proprii.

Domenii de conținut	Conținuturi
	3.11. Clase ale limbajului Python pentru interacțiunea cu bazele de date - caracteristici; - metode ale claselor principale (de exemplu, Connection, Cursor) pentru operații uzuale: conectare la baza de date, executare a unei comenzi, închidere a conexiunii cu baza de date; - repere pentru identificarea și utilizarea altor clase și metode adecvate pentru interacțiunea cu bazele de date, în funcție de nevoile proprii. **3.12. Biblioteci ale limbajului Python pentru învățare automată** - caracteristici, clase și metode specifice din biblioteca Matplotlib pentru vizualizarea datelor: reprezentarea grafică a datelor prin puncte, grafic liniar, diagramă cu bare, histogramă, diagramă box-plot; - caracteristici, clase și metode specifice din biblioteca Pandas pentru curățarea, manipularea și analiza statistică a datelor: tabele (DataFrames), eliminarea valorilor extreme, obținere a valorilor pentru un atribut, statistici descriptive (medie, deviație standard, minim, maxim); - caracteristici, clase și metode specifice din biblioteca Scikit-learn pentru rezolvarea problemelor de codificare și normalizare a datelor, clasificare, regresie și clusterizare: metode pentru antrenare, predicție și evaluare în rezolvarea problemelor de învățare automată; - repere pentru utilizarea algoritmilor prin metodele disponibile în biblioteci, pentru rezolvarea unor probleme de natură practică; - repere pentru identificarea și utilizarea altor clase și metode adecvate pentru învățare automată, în funcție de nevoile proprii.

Anexa nr. 13.2. Programa de examen pentru disciplina *Informatică* – INF_P2 (CS)

- *Proba E – a doua probă (probă scrisă, specifică profilului/specializării)*
- *Filiera teoretică, profilul real, specializarea științe ale naturii*

PREAMBUL

Programa de examen pentru disciplina *Informatică* fundamentează elaborarea variantelor de subiecte și a baremelor de evaluare și de notare pentru **proba E – a doua probă** din cadrul examenului național de bacalaureat 2030. Prezenta programă precizează **competențele de evaluat**, **domeniile de conținut** și **conținuturile** vizate de evaluarea specifică acestui examen.

Programa de examen derivă din **programele școlare în vigoare pentru învățământul liceal**, aprobate prin **Ordinul ministrului educației și cercetării nr. 6930/2025 și Ordinul ministrului educației și cercetării nr. 4370/2026**. Evaluarea vizează competențele specifice care contribuie la formarea/dezvoltarea competențelor generale pe parcursul întregului ciclu liceal, prin raportare la profilul de formare al absolventului.

Obiectivul evaluării prin proba de examen la această disciplină este de a constata măsura în care candidatul *recunoaște, explică, utilizează, analizează, evaluează și creează* în domeniul informatic, în situații de complexitate corespunzătoare raportării (*CS*) și tipului de probă.

Prin susținerea probei de examen la disciplina informatică, candidatul este pus în situația de a face dovada următoarelor **competențe generale (CG)**, dobândite pe parcursul învățământului liceal (clasele a IX-a – a X-a):

CG1	Identifică principalele caracteristici ale modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru înțelegerea fundamentelor programării
CG2	Explică principii care stau la baza modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru a fundamenta în mod logic proiectarea și implementarea soluțiilor informatice
CG3	Utilizează modele conceptuale și operaționale ale dezvoltării produselor software, în scopul obținerii de soluții informatice funcționale și eficiente
CG4	Analizează caracteristicile și aplicabilitatea modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru a selecta soluțiile cele mai potrivite în funcție de contextul informatic dat
CG5	Evaluează corectitudinea și eficiența soluțiilor informatice, în vederea optimizării și asigurării funcționalității în diverse scenarii de utilizare
CG6	Elaborează algoritmi și programe personalizate, pentru a crea soluții informatice coerente și adaptate cerințelor

Prin variantele de subiecte elaborate pentru examenul național de bacalaureat se evaluează competențele și conținuturile din prezenta programă, iar baremele de evaluare și de notare prevăd acordarea punctajelor pentru **orice modalitate corectă de rezolvare** a cerințelor.

COMPETENȚE DE EVALUAT

Fiecărei competențe generale îi sunt asociate **competențele specifice (CS)** care pot fi evaluate prin examen, selectate din programele școlare în vigoare și prezentate cumulativ pentru ciclul liceal:

CG1 – Identifică principalele caracteristici ale modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru înțelegerea fundamentelor programării

Cod CS	Competențe specifice de evaluat
IX.CS1.1.	Identifică principalele caracteristici ale organizării datelor în cadrul unor modele conceptuale fundamentale - date simple sau liste, pentru a structura și accesa date în vederea prelucrării acestora
X.CS1.1.	Identifică principalele caracteristici ale organizării datelor în cadrul unor modele conceptuale simple - liniare, asociative, mixte, pentru a structura și accesa date în vederea prelucrării acestora

Cod CS	Competențe specifice de evaluat
IX.CS1.2.	Identifică specificul, caracteristicile și etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea algoritmică a numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS1.2.	Identifică specificul, caracteristicile și etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea listelor
IX.CS1.3.	Identifică specificul, caracteristicile și etapele unor strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS1.3.	Identifică specificul, caracteristicile și etapele unor strategii de tip Greedy pentru rezolvarea problemelor
IX.CS1.4.	Identifică principalele elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea rezolvării eficiente a problemelor informatice
X.CS1.4.	Identifică principalele elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele simple - liniare, asociative, mixte, în vederea rezolvării eficiente a problemelor informatice
IX.CS1.5.	Identifică elementele de sintaxă și componentele din definiția și apelul subprogramelor și subprogramelor predefinite pentru operații uzuale, pentru a le utiliza în implementarea algoritmilor în limbaj de programare

CG2 – Explică principii care stau la baza modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru a fundamenta în mod logic proiectarea și implementarea soluțiilor informatice

Cod CS	Competențe specifice de evaluat
IX.CS2.1.	Explică principiile de organizare a datelor în cadrul unor modele conceptuale fundamentale - date simple sau liste, pentru a le gestiona eficient
X.CS2.1.	Explică principiile de organizare și prelucrare a datelor în cadrul unor modele conceptuale simple - liniare, asociative, mixte, pentru a le gestiona eficient
IX.CS2.2.	Explică etapele algoritmilor specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea algoritmică a numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS2.2.	Explică etapele unor algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea listelor
IX.CS2.3.	Explică etapele unor strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS2.3.	Explică etapele unor strategii de tip Greedy pentru rezolvarea problemelor
IX.CS2.4.	Explică rolul principalelor elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program în vederea rezolvării eficiente a problemelor informatice
X.CS2.4.	Explică rolul principalelor elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele simple - liniare, asociative, mixte, în vederea rezolvării eficiente a problemelor informatice
IX.CS2.5.	Explică mecanismul de executare a subprogramelor și subprogramelor predefinite pentru operații uzuale, pentru a le utiliza în implementarea algoritmilor în limbaj de programare

CG3 – Utilizează modele conceptuale și operaționale ale dezvoltării produselor software, în scopul obținerii de soluții informatice funcționale și eficiente

Cod CS	Competențe specifice de evaluat
IX.CS3.1.	Utilizează modul de organizare și prelucrare a datelor în cadrul unor modele conceptuale fundamentale - date simple sau liste, pentru a rezolva probleme de gestionare a datelor
X.CS3.1.	Utilizează modul de organizare și prelucrare a datelor în cadrul unor modele conceptuale simple - liniare, mixte, pentru a rezolva probleme de gestionare a datelor

Cod CS	Competențe specifice de evaluat
IX.CS3.2.	Aplică algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea algoritmică a numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS3.2.	Aplică algoritmi specializați pe clase de probleme, pentru a îi putea utiliza în prelucrarea listelor
IX.CS3.3.	Aplică strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS3.3.	Aplică strategii de tip Greedy pentru rezolvarea problemelor
IX.CS3.4.	Utilizează principalele elemente ale limbajului de programare utilizate pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea rezolvării eficiente a problemelor informatice
X.CS3.4.	Utilizează principalele elemente ale limbajului de programare pentru prelucrarea datelor organizate în modele simple - liniare, asociative, mixte, în vederea rezolvării eficiente a problemelor informatice
IX.CS3.5.	Utilizează elementele de sintaxă și componentele din definiția și apelul subprogramelor și subprogramelor predefinite pentru operații uzuale, în implementarea algoritmilor în limbaj de programare

CG4 – Analizează caracteristicile și aplicabilitatea modelelor conceptuale și operaționale ale dezvoltării produselor software, pentru a selecta soluțiile cele mai potrivite în funcție de contextul informatic dat

Cod CS	Competențe specifice de evaluat
IX.CS4.1.	Analizează caracteristicile, modul de prelucrare, avantajele și dezavantajele utilizării unor modele conceptuale fundamentale - date simple sau liste, pentru a alege structura adecvată în rezolvarea unor probleme concrete
X.CS4.1.	Analizează caracteristicile, modul de prelucrare, avantajele și dezavantajele utilizării unor modele conceptuale simple - liniare, asociative, mixte, pentru a alege structura adecvată în rezolvarea unor probleme concrete
IX.CS4.2.	Analizează comportamentul, aplicabilitatea, avantajele și dezavantajele utilizării unor algoritmi specializați pe clase de probleme pentru prelucrarea numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS4.2.	Analizează comportamentul, aplicabilitatea, avantajele și dezavantajele utilizării unor algoritmi specializați pe clase de probleme pentru prelucrarea listelor
IX.CS4.3.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS4.3.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor strategii de tip Greedy pentru rezolvarea problemelor
IX.CS4.4.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor elemente ale limbajului de programare, pentru a identifica soluția adecvată prelucrării datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea rezolvării eficiente a problemelor informatice
X.CS4.4.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării unor elemente ale limbajului de programare, pentru a identifica soluția adecvată prelucrării datelor organizate în modele simple - liniare, asociative, mixte, în vederea rezolvării eficiente a problemelor informatice
IX.CS4.5.	Analizează aplicabilitatea, avantajele și dezavantajele utilizării subprogramelor, în implementarea algoritmilor în limbaj de programare

CG5 – Evaluează corectitudinea și eficiența soluțiilor informatice, în vederea optimizării și asigurării funcționalității în diverse scenarii de utilizare

Cod CS	Competențe specifice de evaluat
IX.CS5.1.	Argumentează alegerea unor modele conceptuale fundamentale - date simple sau liste - și a modurilor de prelucrare a acestora pentru rezolvarea unor probleme informatice concrete

Cod CS	Competențe specifice de evaluat
X.CS5.1.	Argumentează alegerea unor modele conceptuale simple - liniare, asociative, mixte și a modurilor de prelucrare a acestora pentru rezolvarea unor probleme informatice concrete
IX.CS5.2.	Evaluează corectitudinea și eficiența soluțiilor cu algoritmi specializați pe clase de probleme pentru prelucrarea numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS5.2.	Evaluează corectitudinea și eficiența soluțiilor cu algoritmi specializați pe clase de probleme pentru prelucrarea listelor
IX.CS5.3.	Evaluează corectitudinea și eficiența algoritmilor care utilizează strategii de rezolvare a problemelor prin proiectarea modulară a algoritmilor
X.CS5.3.	Evaluează corectitudinea și eficiența algoritmilor care utilizează strategii de tip Greedy pentru rezolvarea problemelor
IX.CS5.4.	Evaluează corectitudinea și eficiența aplicațiilor care utilizează elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea rezolvării eficiente a problemelor informatice
X.CS5.4.	Evaluează corectitudinea și eficiența aplicațiilor care utilizează elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele simple - liniare, mixte, în vederea rezolvării eficiente a problemelor informatice
IX.CS5.5.	Evaluează corectitudinea și eficiența programelor care utilizează subprograme, pentru a rezolva probleme informatice

CG6 – Elaborează algoritmi și programe personalizate, pentru a crea soluții informatice coerente și adaptate cerințelor

Cod CS	Competențe specifice de evaluat
IX.CS6.1.	Proiectează algoritmi personalizați care utilizează modele conceptuale fundamentale - date simple sau liste pentru organizarea și prelucrarea eficientă a datelor, utilizând algoritmi de bază, în vederea rezolvării unor probleme
X.CS6.1.	Proiectează algoritmi personalizați care utilizează modele conceptuale simple - liniare, asociative, mixte, pentru organizarea și prelucrarea eficientă a datelor, utilizând algoritmi de bază, în vederea rezolvării unor probleme
IX.CS6.2.	Proiectează soluții cu aplicarea personalizată a algoritmilor specializați pe clase de probleme pentru prelucrarea numerelor, sortarea sau generarea sistematică a unor secvențe de valori
X.CS6.2.	Proiectează soluții cu aplicarea personalizată a algoritmilor specializați pe clase de probleme pentru prelucrarea listelor
IX.CS6.3.	Proiectează modular algoritmi personalizați pentru rezolvarea problemelor
X.CS6.3.	Proiectează algoritmi cu aplicarea personalizată a strategiilor de tip Greedy pentru rezolvarea problemelor
IX.CS6.4.	Implementează aplicații care integrează în mod personalizat elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele fundamentale - date simple sau liste, respectiv pentru transmiterea datelor de la și către un program, în vederea realizării unor soluții funcționale și performante
X.CS6.4.	Implementează aplicații care integrează în mod personalizat elemente specifice de limbaj de programare pentru prelucrarea datelor organizate în modele simple - liniare, asociative, mixte, în vederea realizării unor soluții funcționale și performante
IX.CS6.5.	Implementează aplicații în limbaj de programare care integrează subprograme personalizate, pentru a modulariza și organiza eficient codul în cadrul soluțiilor informatice

DOMENII DE CONȚINUT ȘI CONȚINUTURI

Conținuturile evaluate sunt organizate pe **domenii de conținut** și sunt prezentate cumulativ pentru ciclul liceal, în acord cu programele școlare în vigoare:

Domenii de conținut	Conținuturi
1. Organizarea conceptuală a datelor	**1.1. Modelul conceptual liniar - listă** - caracteristici, din punctul de vedere conceptual, ale unei liste și ale cazurilor particulare date de tipul de acces (cu acces direct, cu acces secvențial), rolul avut în prelucrarea datelor (listă de frecvențe); - repere pentru parcurgerea elementelor și pentru aplicarea algoritmilor de bază pentru prelucrarea datelor organizate liniar, cu sau fără memorare. **1.2. Modelul conceptual liniar – șir de caractere** - caracteristici și repere pentru acces la date, parcurgerea lor și pentru aplicarea algoritmilor de bază în scopul prelucrării acestora. **1.3. Modelul conceptual mixt (de exemplu, liste de liste, liste de șiruri de caractere)** - caracteristici și repere pentru acces la date, parcurgerea lor și pentru aplicarea algoritmilor de bază sau specifici în scopul prelucrării acestora (de exemplu, actualizarea datelor, modificarea numărului de componente ale colecției, construirea colecției).
2. Strategii de rezolvare a problemelor	**2.1. Principii de elaborare a unui program** - caracteristici ale gândirii computaționale și ale principalelor etape ale elaborării unui program (analiză, proiectare, implementare, testare și depanare), repere pentru aplicarea acestor etape în rezolvarea problemelor; - caracteristici de bază, avantaje și limite ale principalelor moduri de reprezentare a algoritmilor: blocuri grafice, pseudocod, limbaj de programare (cu interpretor sau compilator); - repere pentru proiectarea modulară a algoritmilor; - criterii de elaborare a testelor (pentru validarea datelor și pentru verificarea comportamentului algoritmului), repere pentru urmărirea evoluției valorilor variabilelor în scopul identificării și tratării erorilor; - eficiența algoritmilor: caracteristici ale eficienței din punctul de vedere al spațiului de memorie și din punctul de vedere al timpului de executare; - caracteristici ale unor modalități de bază de comunicare cu programul: interfață de tip consolă, fișiere. **2.2. Prelucrări ale datelor numerice** - operații specifice cu cifrele unui număr (acces la o cifră a unui număr, adăugare a unei cifre la stânga unui număr, adăugare a unei cifre la dreapta unui număr), determinarea unui divizor/multiplu al unui număr; - repere pentru prelucrarea numerelor (de exemplu parcurgerea cifrelor unui număr, a divizorilor unui număr, descompunere a unui număr în factori primi) și aplicarea algoritmilor de bază; - algoritmi elementari pentru determinarea celui mai mare divizor comun (algoritmul lui Euclid cu scăderi repetate, respectiv cu împărțiri repetate). **2.3. Metode de generare sistematică a elementelor unei liste** - repere pentru generarea unor secvențe de valori (de exemplu secvențe cu proprietăți date, termeni ai unor expresii matematice, termeni ai unor șiruri recurente) și aplicarea algoritmilor de bază. **2.4. Metode de sortare a elementelor unei liste** - caracteristici ale metodei de sortare prin selecția minimului și repere pentru aplicarea metodei; - caracteristici ale metodei de sortare cu listă de frecvențe și repere pentru aplicarea metodei. **2.5. Metode de prelucrare a listelor sortate** - caracteristici ale metodei căutării binare și repere pentru aplicarea acesteia; - caracteristici ale metodei interclasării și repere pentru aplicarea acesteia. **2.6. Metoda Greedy** - caracteristici ale metodei Greedy; - repere pentru aplicarea metodei.

Domenii de conținut	Conținuturi
3. Memorarea datelor și organizarea codului în limbaj de programare	**3.1. Subprograme** - caracteristici, rol; - concepte de bază și caracteristici: antet, corp, variabile locale, variabile globale, transmitere prin intermediul parametrilor, returnare a rezultatelor, apel, mecanism de executare; - sintaxă pentru definiția și apelul unui subprogram în Python; - caracteristici și repere de utilizare a unor subprograme predefinite în Python pentru operații matematice uzuale (de exemplu radical, valoare absolută, parte întreagă, rotunjire) și conversii; - caracteristici și repere de utilizare a unor subprograme predefinite în Python pentru colecții de valori (determinarea numărului de elemente, a minimului, maximului, sumei unor valori); **3.2. Introducere în programarea orientată pe obiecte în limbaj de programare** - noțiuni de bază necesare utilizării unor clase predefinite: clasă, membri ai clasei (date și metode), obiecte, biblioteci; - instanțiere a unei clase predefinite, acces la membrii unui obiect. **3.3. Fișiere text** - caracteristici, principii de lucru (deschidere, închidere, transfer de date); - funcție pentru deschiderea unui fișier în Python, clasa predefinită TextIOWrapper din Python pentru prelucrarea fișierelor: caracteristici, metode de bază (pentru citire, scriere, închidere); - repere pentru identificarea și utilizarea altor clase și metode, adecvate pentru prelucrarea fișierelor și pregătirea datelor citite din fișiere. **3.4. Clasa list din Python – clasă predefinită pentru memorarea unei liste** - caracteristici; - operatori specifici pentru acces la un element, apartenență, non-apartenență, concatenare, multiplicare, relaționare; - metode ale clasei pentru operații de bază: determinare a primei poziții a unei valori, numărare a aparițiilor unei valori, ștergere a elementului de la o anumită poziție, inserare a unei valori într-o anumită poziție, adăugare a unui element, copiere, sortare; - repere pentru identificarea și utilizarea altor metode ale clasei, adecvate pentru prelucrarea listelor. **3.5. Clasa str din Python – clasă predefinită pentru șiruri de caractere** - caracteristici; - operații și operatori specifici: apartenență, concatenare, multiplicare, acces la un caracter sau subșir/secvență de caractere, comparare lexicografică; - metode uzuale pentru căutare a unei secvențe de caractere, generare a unui șir prin înlocuire a unei secvențe de caractere, separare a unor secvențe de caractere dintr-un șir; - repere pentru identificarea și utilizarea altor metode și funcții adecvate pentru prelucrarea șirurilor de caractere, în funcție de nevoile proprii. **3.6. Elemente de limbaj pentru operarea cu date organizate în modele mixte** - caracteristici, acces la un element în Python.