PLANIFICARE CALENDARISTICĂ ANUALĂ

Model orientativ pentru manual • An școlar 2026–2027

Disciplina: Informatică • Clasa a IX-a • Curriculum de specialitate (CS), științe ale naturii • 1 oră/săptămână (teorie și aplicații practice integrate)

Model general • 34 de săptămâni de activități • 34 ore

Repartizarea orientativă a orelor

Modul	Capitol / activitate	Ore disciplină	Ore R incluse	Alte activități
I	Gândirea computațională. Strategii de rezolvare a problemelor	5	1	—
II	Prelucrări numerice. Algoritmi elementari	6	1	—
III	Subprograme. Introducere în programarea orientată pe obiecte	7	2	—
IV	Modalități de comunicare cu programul. Fișiere text	4	2	—
V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	12	3	—
TOTAL	5 capitole	34	9	—

Din cele 34 de ore de informatică, 25 de ore sunt destinate predării, aplicării şi evaluării, iar 9 ore sunt rezervă curriculară R (aprox. 25%), pentru consolidare, aprofundare, diferențiere și activități remediale. În săptămânile marcate R: 1 oră R; în celelalte săptămâni: 1 oră pentru predare, aplicații și evaluare.

Intervalele pentru „Săptămâna verde”, „Școala altfel” și vacanța mobilă se stabilesc conform calendarului aplicabil fiecărei unități de învățământ. Repartizarea pe săptămâni este orientativă și se adaptează orarului.

Planificarea materiei pe săptămâni

Modul / perioadă	Unitatea de învățare	Competențe specifice	Conținuturi / activități	Nr. ore	Săptă mâna	Observații
M I	Gândirea computațională. Strategii de rezolvare a problemelor	CS 1.3, 2.3, 3.3	**I.1. Gândirea computațională** • Definiția gândirii computaționale; avantajul aplicării în multiple domenii (educație, cercetare, medicină, economie, industrie) • Cele patru caracteristici: descompunerea, recunoașterea tiparelor, abstractizarea, algoritmizarea • Studiu de caz: semafoare inteligente (mapare pe cele 4 caracteristici) • Activități: analiză de situații cotidiene; proiect de echipă (cele 4 componente pe o situație din viața reală); portofoliu digital	1	S1	1 oră (predare/aplicații/evaluare)
M I	Gândirea computațională. Strategii de rezolvare a problemelor	CS 1.3, 2.3, 4.3	**I.2. Principalele moduri de reprezentare a algoritmilor** • Noțiunea de algoritm; proprietăți: generalitate, corectitudine, finitudine, claritate, eficiență • Exemplu cotidian: împrumutul unei cărți de la bibliotecă (pași ordonați; observarea rolului fiecărui pas) • Blocurile unei scheme logice • Elementele principale ale limbajului pseudocod • Clasificarea limbajelor de programare: nivel scăzut / înalt; compilate / interpretate; • Concluzii (schema logică / pseudocod / limbaj interpretat / compilat) • Activități: reprezentări ale unor algoritmi folosind schema logică și limbajul pseudocod; portofoliu digital	1	S2	1 oră (predare/aplicații/evaluare)
M I	Gândirea computațională. Strategii de rezolvare a problemelor	CS 3.3, 5.3	**I.3. Testarea algoritmilor. Urmărirea evoluției valorilor variabilelor** • De ce testăm: corectitudine, completitudine, robustețe; acoperirea cerințelor • Criterii de testare: cazuri generale; cazuri limită; cazuri speciale; valori invalide/exceptate; acoperirea tuturor ramurilor decizionale • Testarea structurilor repetitive: zero executări / o executare / N executări • Urmărirea evoluției variabilelor (tabele de evoluție a valorilor variabilelor) • Activități: proiectarea unui set de teste pentru un algoritm dat și depanarea sa	1	S3	1 oră (predare/aplicații/evaluare)
M I	Gândirea computațională. Strategii de rezolvare a problemelor	CS 2.3, 3.3, 5.3	**I.4. Principalele etape ale elaborării unui program** • Etape: analiza problemei; proiectarea soluției; implementarea; testarea; depanarea • Descrierea etapelor elaborării unui program • Activități: aplicații	1	S4	1 oră (predare/aplicații/evaluare)
M I	Gândirea computațională. Strategii de rezolvare a problemelor	CS 3.3, 4.3, 5.3	**Probleme propuse și fișă de evaluare — Capitolul I** • Evaluare formativă/sumativă pe competențele CS 1.3–5.3; feedback • Activități R: probleme propuse / fișă de evaluare capitolul I; remediere	1	S5	1 oră R
M II	Prelucrări numerice. Algoritmi elementari	CS 1.2, 2.2, 3.2	**II.1. Accesarea cifrelor unui număr** • Tipuri de date și variabile în Python; operatori aritmetici • Parcurgerea cifrelor unui număr natural •Construirea unor rezultate pe baza cifrelor • Activități: aplicații	1	S6	1 oră (predare/aplicații/evaluare)

M II	Prelucrări numerice. Algoritmi elementari	CS 2.2, 3.2	**II.2. Algoritmi de construire a numerelor prin adăugarea cifrelor la dreapta** • Adăugarea unei cifre; adăugarea succesivă a mai multor cifre • Adăugare condiționată: if; if…else • Operatori relaționali • Activități: aplicații	1	S7	1 oră (predare/aplicații/evaluare)
M II	Prelucrări numerice. Algoritmi elementari	CS 2.2, 3.2	**II.3. Algoritmi de construire a numerelor prin adăugarea cifrelor la stânga** • Ideea păstrării ordinii cifrelor la inserarea în stânga; puteri ale lui 10 • Operatori logici • Structuri repetitive (while) pentru prelucrarea cifrelor; • Aplicații: sumă a cifrelor; variante de construire a unui număr nou • Activități: exerciții cu operatori logici în condiții compuse	1	S8	1 oră (predare/aplicații/evaluare)
M II	Prelucrări numerice. Algoritmi elementari	CS 3.2, 4.2	**II.4. Divizorii unui număr natural** • Noțiunea de divizor; numărul și suma divizorilor • Instrucțiunea for și funcția range (start, stop, pas) — reguli și greșeli frecvente • Testarea primalității; aplicații pe divizori • Activități: probleme de numărare/calculul sumei divizorilor; optimizări elementare de interval	1	S9	1 oră (predare/aplicații/evaluare)
M II	Prelucrări numerice. Algoritmi elementari	CS 3.2, 6.2	**II.5. Algoritmi elementari pentru divizori și multipli comuni** • Descompunerea în factori primi • CMMDC • algoritmi elementari pentru determinarea celui mai mare divizor comun (algoritmul lui Euclid cu scăderi repetate, respectiv cu împărțiri repetate) • CMMMC; relația cu CMMDC; multiplii unui număr • Activități: aplicații combinate (factori primi + CMMDC/CMMMC)	1	S10	1 oră (predare/aplicații/evaluare)
M II	Prelucrări numerice. Algoritmi elementari	CS 3.2, 4.2, 6.2	**Probleme propuse și fișă de evaluare — Capitolul II** • Aplicații integrate pe prelucrări numerice (cifre, divizori) • Fișă de evaluare de capitol; barem / feedback • Activități R: remediere pe itemii cu erori frecvente; aprofundare selectivă • Evaluare formativă/sumativă	1	S11	1 oră R
M III	Subprograme. Introducere în programarea orientată pe obiecte	CS 1.5, 2.5, 3.3, 3.5, 6.3	**III.1. Proiectarea modulară a algoritmilor. Funcții fără parametri** • De ce modularizăm: reutilizare, depanare, lucru în echipă • Definirea și apelul unei funcții fără parametri; instrucțiunea return • Diferența dintre funcții care afișează și funcții care returnează o valoare • Activități: funcții simple (par/impar, oglindit, nr. cifre, sumă divizori)	1	S12	1 oră (predare/aplicații/evaluare)
M III	Subprograme. Introducere în programarea orientată pe obiecte	CS 1.5, 2.5, 3.5	**III.2. Funcții cu parametri (partea I)** • Antetul funcției; parametri formali vs. parametri efectivi • Valori returnate • Aplicații • Activități: scriere și testare de funcții	1	S13	1 oră (predare/aplicații/evaluare)
M III	Subprograme. Introducere în programarea orientată pe obiecte	CS 2.5, 4.5, 6.5	**III.2. Variabile locale și globale; mecanismul apelului (partea II)** • Vizibilitatea variabilelor: locale vs. globale • Parametri cu valori implicite	1	S14	1 oră (predare/aplicații/evaluare)

			• Mecanismul de executare al unei funcții • Activități: analiza secvențelor; corectarea erorilor de vizibilitate			
M III	Subprograme. Introducere în programarea orientată pe obiecte	CS 3.5, 4.5	**III.3. Module. Funcții predefinite** • import al unui modul; import selectiv; apel modul.funcție • Module și funcții predefinite utile (math) • 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 • Activități: mini-aplicații cu import; documentare minimală a unei funcții	1	S15	1 oră (predare/aplicații/evaluare)
M III	Subprograme. Introducere în programarea orientată pe obiecte	CS 1.5, 2.5, 4.5	**III.4. Introducere în programarea orientată pe obiecte** • Clasă, obiect, instanță; atribute și metode (exemple: Elev, int, Fraction) • Crearea și utilizarea obiectelor; metode tipice • Biblioteci / tipuri predefinite ca exemple de clase • Activități: explorare Fraction / float; exerciții de identificare clasă vs. obiect	1	S16	1 oră (predare/aplicații/evaluare)
M III	Subprograme. Introducere în programarea orientată pe obiecte	CS 3.5, 6.3, 6.5	**Probleme propuse — Capitolul III** • Aplicații modulare (funcții + module + elemente OOP introductive) • Activități R: proiect scurt modular	1	S17	1 oră R
M III	Subprograme. Introducere în programarea orientată pe obiecte	CS 3.5, 5.5, 6.3, 6.5	**Fișă de evaluare — Capitolul III** • Evaluare formativă/sumativă • Activități R: remediere	1	S18	1 oră R
M IV	Modalități de comunicare cu programul. Fișiere text	CS 1.4, 2.4, 3.4	**IV.1. Gestionarea comunicării cu utilizatorul. Principii de lucru cu fișiere text** • Comunicare prin consolă vs. fișiere vs. interfață • Utilitatea fișierelor text; principii de bază; clasa TextIOWrapper • Deschidere/închidere; moduri de acces; • Citire/scriere elementară • Activități: exerciții open/read/write	1	S19	1 oră (predare/aplicații/evaluare)
M IV	Modalități de comunicare cu programul. Fișiere text	CS 3.4, 4.4, 6.4	**IV.2. Aplicații practice cu fișiere text** • Programe complete: citire din fișier, prelucrare, scriere rezultate • Exemplu tip Fibonacci scris în fișier; variabile de stare • Activități: aplicații	1	S20	1 oră (predare/aplicații/evaluare)
M IV	Modalități de comunicare cu programul. Fișiere text	CS 3.4, 5.4, 6.4	**IV.2. Probleme cu fișiere text (continuare / aprofundare)** • Prelucrări numerice din fișier; validare; fișiere de intrare/ieșire • Probleme tip: medii elevi, încasări pe zile, curs valutar • Activități R: depanare pe fișiere reale de test; diferențiere • Evaluare formativă/sumativă	1	S21	1 oră R
M IV	Modalități de comunicare cu programul. Fișiere text	CS 3.4, 5.4, 6.4	**Probleme propuse și fișă de evaluare — Capitolul IV** • Miniaplicații integrate (fișiere) • Evaluare formativă/sumativă • Activități R: proiect	1	S22	1 oră R

M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 1.1, 2.1, 3.1, 5.1	**V.1. Modelul conceptual liniar** • Definiția și caracteristicile unei liste; ordine, lungime, acces • Acces direct vs. acces secvențial; parcurgere • Lista de frecvențe — utilitate pe domenii restrânse de valori • repere pentru parcurgerea elementelor și pentru aplicarea algoritmilor de bază pentru prelucrarea datelor organizate liniar, cu sau fără memorare • Activități: aplicații	1	S23	1 oră (predare/aplicații/evaluare)
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 1.4, 2.4, 3.4	**V.2. Clasa list. Accesarea elementelor. Operatori specifici** • Caracteristicile tipului list în Python; creare prin constante • Indexare; slice-uri; operatori (apartenență, concatenare, multiplicare, relaționali) • Greșeli frecvente la indexare / slice-uri • Activități: aplicații	1	S24	1 oră (predare/aplicații/evaluare)
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 3.1, 3.5, 6.1	**V.3. Crearea unei liste prin citire. Funcții predefinite pentru liste** • Citirea a n elemente într-o listă; parcurgeri • Funcții predefinite: len, min, max, sum — comportament și utilizare • Activități: aplicații	1	S25	1 oră (predare/aplicații/evaluare)
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 3.2, 3.4, 6.4	**V.4. Metode ale clasei list pentru adăugarea, inserarea, ștergerea și căutarea unui element** • Adăugare / inserare / ștergere / căutare: append, insert, remove, pop, index, count • sort / sorted — diferențe (in-place vs. listă nouă) • Activități: construirea listelor pas cu pas; căutări și actualizări	1	S26	1 oră (predare/aplicații/evaluare)
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 3.2, 3.4, 6.4	**V.5. Alte metode ale clasei list. Generarea sistematică a elementelor unei liste** • Alte metode: copy, sort, reverse, extend, clear • Generarea sistematică a elementelor • Activități: aplicații	1	S27	1 oră (predare/aplicații/evaluare)
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 3.2, 3.4, 6.4	**V.4–V.5. Metode ale clasei list; generare sistematică** • Adăugare / inserare / ștergere / căutare: append, insert, remove, pop, index, count, copy, etc • Generarea sistematică a elementelor • Activități R: construirea listelor pas cu pas; căutări și actualizări	1	S28	1 oră R
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 3.2, 4.2, 5.2, 6.2	**V.6. Metode de sortare a elementelor unei liste** • Sortarea prin selecție • Activități: aplicații practice	1	S29	1 oră (predare/aplicații/evaluare)
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 3.1, 5.2, 6.1	**V.7. Liste de frecvențe. Sortarea cu listă de frecvențe** • Construirea unei liste de frecvențe pe un domeniu restrâns • Sortare / reordonare pe baza frecvențelor • Activități: probleme de numărare și afișare după frecvență; consolidare	1	S30	1 oră (predare/aplicații/evaluare)
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 4.1, 5.2, 5.3, 6.1	**V.8. Eficiența algoritmilor** • Ideea de ordine de complexitate	1	S31	1 oră (predare/aplicații/evaluare)

			• 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)			
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 4.1, 5.2, 5.3, 6.1	**V.8. Eficiența algoritmilor (continuare)** • Compararea unor soluții pe același tip de problemă • Activități R: consolidare	1	S32	1 oră R
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 4.1, 5.2, 6.1	**Probleme propuse și fișă de evaluare — Capitolul V** • Fișă de evaluare de capitol; barem / feedback • Activități R: remediere pe itemii cu erori frecvente; aprofundare selectivă	1	S33	1 oră R
M V	Modelul conceptual liniar. Clasa list. Metode de sortare. Eficiența algoritmilor	CS 5.1, 5.2, 5.3, 5.5	**Sinteză anuală. Evaluare finală** • Sinteză anuală • Evaluare finală și feedback	1	S34	1 oră (predare/aplicații/evaluare)

R = activitate din rezerva curriculară (consolidare, aprofundare, recuperarea lacunelor, investigație, proiect). Orele de teorie și aplicațiile practice se integrează în ora săptămânală. Numerotarea S1–S34 este orientativă.