

Planificare calendaristică Anul școlar 2025-2026

Programa aprobată cu O.M. Nr.5099/09.09.2009
Filiera teoretică, toate profilurile și specializările

COMPETENȚE SPECIFICE :

1. Identificarea conexiunilor dintre informatică și societate
 - 1.1. Identificarea aplicațiilor informaticii în viața socială
 - 1.2. Recunoașterea situațiilor în care este necesară prelucrarea algoritmică a informațiilor.
2. Identificarea datelor care intervin într-o problemă și a relațiilor dintre acestea
 - 2.1. Descompunerea rezolvării unei probleme în pași
 - 2.2. Identificarea tipurilor de date necesare pentru rezolvarea unei probleme (de intrare, de ieșire, de manevră).
 - 2.3. Descrierea coerentă a unei succesiuni de operații prin care se obțin din datele de intrare, datele de ieșire
3. Elaborarea algoritmilor de rezolvare a problemelor
 - 3.1. Analizarea enunțului unei probleme și stabilirea pașilor de rezolvare a problemei.
 - 3.2. Reprezentarea algoritmilor în pseudocod.
 - 3.3. Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.
4. Implementarea algoritmilor într-un limbaj de programare
 - 4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării
 - 4.2. Alegerea unui algoritm eficient de rezolvare a unei probleme
5. Aplicarea algoritmilor fundamentali în prelucrarea datelor
 - 5.1. Identificarea într-un program a structurilor de control învățate

Unitatea de învățare	Competențe specifice	Conținuturi	Nr. ore alocate	Săptămâna	Observații
MODULUL 1					
Identificarea conexiunilor dintre informatică și societate	1.1 Identificarea aplicațiilor informaticii în viața socială. 1.2. Recunoașterea situațiilor în care este necesară prelucrarea algoritmică a informațiilor.	- Definirea informaticii ca știință - Rolul informaticii în societate - Studii de caz ale unor situații sociale, în abordare informatizată	2	S1-S2	

Identificarea datelor care intervin într-o problemă și a relațiilor dintre acestea	2.1. Descompunerea rezolvării unei probleme în pași. 2.2. Identificarea tipurilor de date necesare pentru rezolvarea unei probleme (date de intrare, de ieșire, de manevră) 2.3. Descrierea coerentă a unei succesiuni de operații care se obțin din datele de intrare, datele de ieșire.	• Etapele rezolvării problemelor. Exemple. • Noțiunea de algoritm. Caracteristici. Exemple. • Date cu care lucrează algoritmi (constante, variabile, expresii). • Operații asupra datelor (aritmetice, logice, relaționale).	5	S3-S7
MODULUL 2				
Elaborarea algoritmilor de rezolvare a problemelor	3.1. Analizarea enunțului unei probleme: identificarea datele de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei. 3.2. Reprezentarea algoritmilor în pseudocod. 3.3. Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.	• Reprezentarea algoritmilor. Pseudocod. • Principiile programării structurate. Structuri de bază: ○ structura liniară ○ structura alternativă ○ structura repetitivă	8	S8-14
MODULUL 3				
Elaborarea algoritmilor de rezolvare a problemelor	3.1. Analizarea enunțului unei probleme: identificarea datele de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei. 3.2. Reprezentarea algoritmilor în pseudocod. 3.3. Respectarea principiilor programării structurate în procesul de elaborare a algoritmilor.	• Algoritmi elementari • Prelucrarea numerelor : ○ prelucrarea cifrelor unui număr (de exemplu, suma cifrelor, testarea proprietății de palindrom, etc.) ○ probleme de divizibilitate (de exemplu, determinarea divizorilor unui număr, determinarea c.m.m.d.c./c.m.m.m.c., testare primalitate, etc.) ○ calculul unor expresii simple (sume, produse, etc.)	7	S15-S21
MODULUL 4				
Elaborarea algoritmilor de rezolvare a problemelor	3.1. Analizarea enunțului unei probleme: identificarea datele de intrare și a datele de ieșire (cu specificarea tipul datelor și a relațiilor existente între date) și stabilirea pașilor de rezolvare a problemei. 3.2. Reprezentarea algoritmilor în pseudocod. 3.3. Respectarea principiilor programării	• Prelucrarea unor secvențe de valori • determinare minim/maxim • verificarea unei proprietăți (de exemplu, dacă toate elementele din secvență sunt numere perfecte, etc.) • calculul unor expresii în care intervin valori din secvență (de exemplu:	6	S22-S27

	structurate în procesul de elaborare a algoritmilor.	numărarea elementelor pare/impare, etc) • generarea șirurilor recurente (de exemplu: șirul Fibonacci)			
MODULUL 5					
Implementarea algoritmilor într-un limbaj de programare	4.1. Elaborarea unui algoritm de rezolvare a unor probleme din aria curriculară a specializării 4.2. Alegerea unui algoritm eficient de rezolvare a unei probleme	• Aplicații interdisciplinare (specifice profilului). • Exemple orientative: • Rezolvarea ecuației de gradul I și de gradul al II-lea • Simplificarea fracțiilor • Aplicații geometrice (distanța dintre două puncte, aria/perimetrul unui triunghi, volumul corpurilor regulate, etc.) • Determinarea punctului de intersecție a două mobile în mișcare rectilinie și uniformă • Determinarea masei moleculare a unui compus chimic. • Analiza eficienței unui algoritm.	6	S28-S33	
Aplicarea algoritmilor fundamentali în prelucrarea datelor	5.1. Identificarea într-un program a structurilor de control învățate	• Exemplificări de implementare a unor algoritmi studiați	3	S34-S36	