

FIȘA DISCIPLINEI

1. Date despre program

Facultatea	Facultatea de Inginerie Electrică și Știința Calculatoarelor
Departamentul	Departamentul de Calculatoare
Domeniul de studii	Calculatoare și tehnologia informației
Ciclul de studii	Licență
Programul de studii	Calculatoare

2. Date despre disciplină

Denumirea disciplinei	INGINERIA PROGRAMELOR				
Anul de studiu	IV	Semestrul	7	Tipul de evaluare	E
Regimul disciplinei	Categoría formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară				DF
	Categoría de opționalitate a disciplinei: DOB – obligatorie, DOP – opțională, DFA - facultativă				DOB

3. Timpul total estimat (ore alocate activităților didactice)

I a) Număr de ore pe săptămână	5	Curs	3	Seminar		Laborator	2	Proiect	
I b) Totalul de ore pe semestru din planul de învățământ	70	Curs	42	Seminar		Laborator	28	Proiect	

II Distribuția fondului de timp pe semestru:	ore
II.a) Studiu individual	77
II.b) Tutoriat (pentru ID)	
III. Examinări	3
IV. Alte activități (precizați):	-

Total ore studiu individual (II.a+II.b+III)	80
Total ore pe semestru (I.b+II.a+II.b+III+IV)	150
Numărul de credite	6

4. Competențe specifice acumulate

Competențe profesionale	CP01. analizează specificații software CP03. definește arhitectura software CP05. dezvoltă prototipul pentru software CP06. proiectează sistemul informatic CP07. transpune cerințele într-un model vizual CP11. realizează schițe de proiectare CP12. utilizează biblioteci de software CP15. identifică cerințele clienților CP16. utilizează instrumente de inginerie software asistată de calculator
Competențe transversale	CT1. munca în echipă

5. Rezultatele învățării

Cunoștințe	Aptitudini	Responsabilitate și autonomie
Studentul/ absolventul trebuie:	Studentul/ absolventul trebuie:	La finalul disciplinei, studenții vor demonstra:
1. Să analizeze critic principiile, etapele și terminologia utilizate în ingineria programelor.	1. Să elaboreze specificații de cerințe software corecte și complete, utilizând tehnici și instrumente adecvate.	• Atitudine critică și responsabilă față de calitatea codului și a procesului de dezvoltare.
2. Să compare principalele metodologii de dezvoltare software și să examineze modul de aplicare a conceptelor de management al proiectelor software.	2. Să construiască modele software prin diagrame UML care să descrie structura și comportamentul sistemului.	• Capacitatea de a lucra în echipă interdisciplinară, de a comunica eficient și de a documenta activitățile desfășurate.
3. Să explice procesul de inginerie a cerințelor și să adapteze conceptele	3. Să proiecteze arhitectura unui sistem software aplicând principii	• Deschidere către învățare

fundamentale ale modelării software prin limbajul UML.	4.	de modularitate și reutilizare. Să implementeze componente software conforme cerințelor definite, folosind instrumente moderne de dezvoltare.	continuă și adaptare la noi tehnologii, metodologii și instrumente software.
4. Să identifice și să aplice principii de proiectare, reutilizare și șabloane de proiectare software adecvate contextului unui proiect.	5.	Să aplice tehnici și instrumente de testare pentru verificarea și validarea produselor software.	• Conștientizarea importanței eticii și securității în dezvoltarea de produse software destinate utilizatorilor reali.
5. Să analizeze critic aspectele legate de testare, calitate și securitate în ciclul de viață al produselor software.			

6. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

Obiectivul general al disciplinei	Studentul/Absolventul trebuie să fie capabil să analizeze, să proiecteze și să implementeze soluții software complexe, de calitate, în conformitate cu cerințele considerate, utilizând metodologii adecvate și instrumente moderne, și să își asume responsabilitatea pentru calitatea, testarea și securitatea produselor program dezvoltate în echipă.
-----------------------------------	---

7. Conținutul predării și învățării

7. Conținutul predării și învățării			
Curs	Nr. ore	Metode de predare	Observații
1. Introducere, definiție, istoric, principii de bază, fazele ingineriei programării. Dicționar termeni	3h	expunerea, prelegerea, conversație, studiu de caz, demonstrația	
2. Metodologii de dezvoltare software	3h		
3. Managementul proiectelor software	2h		
4. Ingineria cerințelor	3h		
5. Modelarea sistemelor software			
5.1. Introducere, limbaje de modelare	2h		
5.2. UML. Diagrame structurale. Exemple	3h		
5.3. UML. Diagrame comportamentale. Exemple	3h		
6. Proiectarea sistemelor software	3h		
7. Implementarea sistemelor software			
7.1. Prezentare generală, ghiduri	2h		
7.2. Instrumente software	2h		
7.3. Reutilizare software	2h		
8. Șabloane de proiectare	3h		
9. Verificare și validare. Testare software. Tipuri de testare. Instrumente pentru managementul testării	3h		
10. Calitatea produselor software	2h		
11. Securitatea sistemelor software. Securitatea în ciclul de dezvoltare a unui produs software	3h		
12. Inteligența artificială generativă și ingineria software	3h		
Bibliografie minimală recomandată			
1. I. Sommerville, <i>Software Engineering</i> , 10th Edition, Pearson, India, 2017			
2. M. Richards and N. Ford, <i>Fundamentals of software architecture: An engineering approach</i> . Sebastopol, CA: O'Reilly Media, 2020.			

3. D. Farley, *Modern software engineering: Doing what works to build better software faster*. Boston, MA: Addison Wesley, 2022.
4. Lara, Letaw. *Handbook of Software Engineering Methods*. 2024.
5. *, Software Engineering Body of Knowledge (SWEBOK), IEEE Computer Society, 2024, <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
6. Suportul electronic de curs
7. Resurse online: tutoriale, articole de specialitate

Aplicații (laborator)	Nr. ore	Metode de predare	Observații
1. Analiza stadiului actual în domeniul proiectului de diplomă/ Identificare și documentare tematică pentru a obține minim 2 direcții relevante pentru tema propusă. Familiarizarea cu un instrument specific de management al proiectelor	2	lucrări practice, exerciții, studii de caz, evaluare	
2. Elaborarea specificațiilor cerințelor software pentru proiectul de diplomă	2		
3. Proiectarea interfeței utilizator	2		
4. Proiectarea aplicației. Realizarea diagramelor UML	2		
5. Proiectare API REST. Proiectare arhitectură II (diagrame UML: secvențe, stări, clase).	2		
6. Stabilire temă comună și echipe de lucru. Elaborarea specificațiilor cerințelor software pentru aplicația de realizat în echipă	2		
7. Proiectare wireframe UI. Proiectare Prototip UI.	2		
8. Proiectarea arhitectura. Realizarea diagramelor UML	2		
9. Planificare implementare versiunea MVP	2		
10. Proiectare baza de date. Implementarea bazei de date. Populare bază de date	2		
11. Implementare schelet aplicație (UI minim integrat cu Backend minim). Dezvoltare module aplicație.	2		
12. Elaborare plan de testare. Testare aplicație. Rafinare implementare. Pregătire aplicație pentru beneficiar	2		
13. Implementarea și execuția unui test automat pentru UI, pentru o aplicație oarecare. Testarea aplicației de către altă echipă	2		
14. Prezentare și evaluare aplicație și documentație elaborată	2		

Bibliografie minimală recomandată

1. Resurse online: tutoriale, materiale bibliografice specifice limbajelor de programare utilizate, articole de specialitate
2. I.Sommerville, *Software Engineering*, 10th Edition, Pearson India, 2017
3. A. Cernian, A.D. Ioniță, *Noțiuni aplicative de inginerie a sistemelor de programe*, Ed. Matrixrom, București, 2009
4. O. Gherghieș, A. Apetrei, *Ingineria programării*, Curs, Iași, 2002
5. I. Odagescu, F. Furtuna, *Ingineria programării*, Ed. ASE, București, 2002
6. Fabian C., Mihalca R., Chichernea V., Goron S., Botezatu C., Iacob I., *Proiectarea sistemelor informatice. Metode de realizare*, Sylvi, Bucuresti 2001
7. D. Bocu –*Inițiere în ingineria sistemelor soft*, Ed. Albastră, Cluj-Napoca, 2001
8. Suportul electronic de laborator

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Înțelegerea critică a teoriilor și principiilor specifice. Capacitatea de utilizare adecvată a termenilor și principiilor din domeniul ingineriei software. Claritatea prezentării aplicației finale și corectitudinea răspunsurilor la întrebări.	Evaluarea cunoștințelor teoretice și practice din tematica studiată în timpul semestrului prin test grilă Evaluarea finală a aplicației practice elaborate.	30 20
Seminar			

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Laborator/ Lucrări practice	Evaluarea capacității de analiză, sinteză și concretizare a cunoștințelor teoretice în identificarea soluțiilor problemei propuse la lucrările de laborator Corectitudinea și calitatea dezvoltării aplicației software și documentației aferente. Corectitudinea planificării, a utilizării instrumentelor, contribuția în echipă	Evaluare continuă a activităților desfășurate	50
Proiect			

Fișa disciplinei include, dacă este cazul, elemente adaptate persoanelor cu dizabilități, în funcție de tipul și gradul acestora.

Data completării	Grad didactic, nume, prenume, semnătura titularului de curs	Grad didactic, nume, prenume, semnătura titularului de aplicație
25.09.2025	Prof.dr.ing. Cristina TURCU	Ș.l.dr.ing. Felicia GÎZĂ

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
25.09.2025	Prof. univ. dr. ing. Cristina Elena TURCU

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
25.09.2025	Prof. univ. dr. ing. Ovidiu-Andrei SCHIPOR

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
26.09.2025	Prof. univ. dr. ing. Laurentiu- Dan MILICI