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	Categoria formativă a disciplinei DF - fundamentală, DS - de specializare, DC – complementară				DF
	Categoria 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	

Distribuția fondului de timp pe semestru:	ore
II. Studiu individual	77
III. Examinări	3
IV. Alte activități (precizați):	

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

4. Competențe specifice acumulate

Competențe transversale	CT1. munca în echipă
Competențe profesionale	CP01. analizează specificații software CP03. definește arhitectura software CP05. dezvolta 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 asistata de calculator

5. Rezultatele învățării

La finalizarea cu succes a disciplinei, studentul va fi capabil să demonstreze următoarele rezultate ale învățării.

Cunoștințe	Aptitudini	Responsabilitate și autonomie
C1. Explică principiile, procesele și modelele fundamentale ale ciclului de viață software. C2. Analizează principiile ingineriei cerințelor, modelării UML, proiectării și arhitecturii software. C3. Explică principiile verificării, validării, testării, securității și asigurării	A1. Identifică cerințele software pornind de la nevoile, obiectivele și constrângerile stakeholderilor. A2. Analizează și elaborează specificații software verificabile pe baza cerințelor identificate. A3. Transpune cerințele software în modele UML și artefacte vizuale care descriu structura și comportamentul sistemului.	RA1. Gestionează activitățile proprii din cadrul unui proiect software în raport cu responsabilitățile, termenele și criteriile de calitate stabilite. RA2. Coordonează activități ale echipei în cadrul responsabilităților asumate. RA3. Evaluează critic rezultatele

calității software. C4. Analizează practicile și instrumentele utilizate pentru dezvoltare colaborativă, managementul configurației, CI/CD și evoluția sistemelor software.	A4. Definește arhitectura unei soluții software pe baza cerințelor, atributelor de calitate și constrângerilor identificate. A5. Dezvoltă incremental un prototip funcțional al soluției software conform cerințelor și arhitecturii stabilite. A6. Selectează și utilizează biblioteci, framework-uri și componente software adecvate proiectului. A7. Utilizează instrumente specializate pentru modelare, managementul proiectului, controlul versiunilor, testare și automatizarea procesului de dezvoltare. A8. Verifică soluția software prin testare, code review și instrumente automate de asigurare a calității.	proprii și ale echipei pe baza cerințelor, testelor și feedbackului primit. RA4. Își asumă responsabilitatea pentru deciziile tehnice, contribuțiile proprii și utilizarea transparentă a instrumentelor digitale și GenAI.

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

Obiectivul general al disciplinei	● Dezvoltarea capacității studenților de a aborda sistematic și integrat ciclul de viață al unui produs software, de la identificarea și analiza cerințelor până la modelare, proiectare arhitecturală, realizarea prototipului, verificare, livrare și evaluare, utilizând metode, modele, instrumente și practici specifice ingineriei software, în contextul muncii colaborative în echipă și al asumării responsabilității pentru deciziile tehnice și calitatea soluției realizate.

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	3h		
4. Ingineria cerințelor	3h		
5. Modelarea sistemelor software			
5.1. Introducere, limbaje de modelare	2h		
5.2. UML. Diagrame structurale. Exemple	2h		
5.3. UML. Diagrame comportamentale. Exemple	2h		
6. Proiectarea sistemelor software	3h		
7. Implementarea sistemelor software	3h		
8. Șabloane de proiectare	3h		
9. Testare software și asigurarea calității	3h		
10. Securitatea sistemelor software. Securitatea in ciclul de dezvoltare a unui produs software	3h		
11. DevOps și CI/CD	3h		
12. Evoluția software. Mentenanță.	3h		

13. Inteligența artificială generativă și ingineria software	3h		
	42		

Bibliografie minimală recomandată

1. I. Sommerville, *Software Engineering*, 10th Edition, Pearson, India, 2017
2. M. Richards and N. Ford, *Fundamentals of software architecture: An engineering approach*. 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 aplicației propuse. Familiarizarea cu un instrument specific de management al proiectelor	2	lucrări practice, exerciții, studii de caz, evaluare	
2. Discuții cu beneficiarul. Elaborarea specificațiilor cerințelor software pentru aplicatie	2		
3. Proiectare arhitectură – etapa I - Diagramele de cazuri de utilizare și activități. Proiectare wireframe UI	2		
4. Proiectare arhitectură – etapa a II-a - (diagrame UML: secvențe, stări, clase, componente). Proiectare Prototip UI	2		
5. Proiectare API REST. Alegerea tehnologiilor pentru backend și frontend. Proiectarea si implementarea bazei de date. Planificare implementare versiunea MVP – task-uri Jira	2		
6. Definire coding standard backend și frontend. Implementare schelet aplicație (UI minimal + Backend minial)	2		
7. Implementare versiune MVP a aplicației. Primirea și adresarea unei recenzii (code review) pe Github.	2		
8. Integrare aplicație MVP (UI + Backend). Documentarea și instalarea aplicațiilor Frontend și Backend. Planificare implementare versiunea 1.0	2		
9. Elaborare plan de testare. Realizarea unor teste de penetrare experimentale. Implementarea și execuția unui test automat pentru UI.	2		
10. Testarea manuală a aplicației MVP. Implementare aplicație 1.0 (UI + Backend).	2		
11. Execuție teste automate si fixare bug-uri. Elaborarea manualului de utilizare	2		
12. Configurarea și testarea aplicației altei echipe.	2		
13. Testare aplicație. Rafinare implementare. Pregătire aplicație pentru beneficiar	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 aplicate de inginerie a sistemelor de programe*, Ed. Matrixrom, București, 2009
4. O. Gherghieș, A. Apetrei, *Ingineria programării*, Curs, Iași, 2002
5. Robert Martin, *Agile Software Development, Principles, Patterns, and Practices,* Prentice Hall, 2002
6. Titus Winters, Tom Manshreck, Hyrum Wright, *Software Engineering at Google,* O'Reilly Media, Inc., 2020
7. Suportul electronic de laborator

8. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Evaluare finală	Cunoașterea și înțelegerea critică a conceptelor, teoriilor și principiilor specifice ingineriei software; utilizarea corectă a terminologiei de specialitate.	Evaluarea cunoștințelor teoretice și practice din tematica studiată în timpul semestrului prin test grilă	30
	Capacitatea de analiză și aplicare a principiilor din domeniul ingineriei software în practică; Capacitatea de utilizare adecvată a termenilor din domeniu; Gradul de îndeplinire a cerințelor; Coerența dintre cerințe, modele, arhitectură și implementare; Calitatea soluției software; Testarea și documentarea; Justificarea deciziilor tehnice; Claritatea prezentării și corectitudinea răspunsurilor la întrebări.	Demonstrarea și susținerea aplicației dezvoltate în echipă; analizarea artefactelor proiectului și întrebări individuale adresate membrilor echipei	20
Evaluare pe parcurs	Capacitatea de analiză, sinteză și aplicare a cunoștințelor în rezolvarea sarcinilor de laborator; Corectitudinea și calitatea artefactelor software și a documentației aferente; Coerența și trasabilitatea cerințelor, Modele, implementare și teste; Utilizarea adecvată a instrumentelor de inginerie software; Planificarea și urmărirea activităților; Contribuția individuală și colaborarea în echipă; Respectarea termenelor și feedbackului primit.	Evaluare continuă a activităților desfășurate la laborator, a livrabilelor intermediare și a contribuției individuale; peer review, code review, verificarea activității în Jira/Confluence și repository.	50

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
23.09.2026	Prof.dr.ing. Cristina TURCU	Ș.l.dr.ing. Felicia GÎZĂ

Data avizării	Grad didactic, nume, prenume, semnătura responsabilului de program
24.09.2026	Prof. univ. dr. ing. Cristina Elena TURCU

Data avizării în departament	Grad didactic, nume, prenume, semnătura directorului de departament
24.09.2026	Prof. univ. dr. ing. Ovidiu-Andrei SCHIPOR

Data aprobării în consiliul facultății	Grad didactic, nume, prenume, semnătura decanului
25.09.2026	Prof. univ. dr. ing. Laurentiu- Dan MILICI