

# FIȘA DISCIPLINEI

## 1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiești
1.2. Facultatea	Inginerie Mecanică și Electrică
1.3. Departamentul	Automatică, Calculatoare și Electronică
1.4. Domeniul de studii universitare	Calculatoare și Tehnologia Informației
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Calculatoare

## 2. Date despre disciplină

2.1. Denumirea disciplinei	<b>Inginerie software</b>
2.2. Titularul activităților de curs	Conf. dr. ing. PRICOP Emil
2.3. Titularul activităților seminar/laborator	Conf. dr. ing. PRICOP Emil
2.4. Titularul activității proiect	-
2.5. Anul de studiu	III
2.6. Semestrul *	2
2.7. Tipul de evaluare	Examen
2.8. Categoria formativă** / regimul*** disciplinei	DD/DOB

\* numărul semestrului este conform planului de învățământ;

\*\* DF - Discipline fundamentale; DS - discipline de specializare; DC - discipline complementare

\*\*\* obligatorie/impusă = DOB; opțională = DOP; facultativă = DFA

## 3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	3	3.3. Seminar/laborator	2	3.4. Proiect	-
3.5. Total ore din planul de învățământ	70	din care: 3.6. curs	42	3.7. Seminar/laborator	28	3.8. Proiect	-
3.9. Total ore studiu individual (studiu după suport de curs, bibliografie și notițe, documentare suplimentară în bibliotecă, pe platformele electronice de specialitate, pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri)							55
3.10. Total ore pe semestru							125
3.11. Numărul de credite							5

## 4. Condiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none"> <li>➤ Programarea calculatoarelor și limbaje de programare</li> <li>➤ Proiectarea algoritmilor</li> <li>➤ Programare orientată pe obiecte</li> <li>➤ Instrumente pentru dezvoltarea programelor</li> </ul>
4.2. de desfășurare a cursului	<ul style="list-style-type: none"> <li>➤ Sală cu dotări multimedia (proiector / display smart)</li> </ul>
4.3. de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> <li>➤ Laborator dotat cu sisteme PC în configurație standard pentru Windows sau Linux, acces la platforma de e-learning a F-IME;</li> <li>➤ Conexiune Internet (utilizare editor de diagrame on-line (draw.io)) sau existența unuia dintre programele Diagrams.NET, Visual Paradigm sau Microsoft Visio;</li> </ul>

## 5. Competențe specifice acumulate și rezultatele învățării\* care stau la baza acestora

Competențe profesionale	Rezultatele învățării*
1. Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor	<b>C1</b> – Studentul/absolventul descrie, identifică și sintetizează concepte și metode fundamentale de inginerie software, referitoare la ciclul de viață al produsului software, modele de proces (cascadă, incremental, agile), gestionarea cerințelor, rolul artefactelor specifice (cerințe, diagrame UML, cod, teste, documentație) în soluționarea problemelor concrete.

	<p><b>A1</b> – Studentul/absolventul elaborează specificații și modele pentru sisteme software de complexitate redusă, folosind tehnici și instrumente specifice ingineriei software (modelare UML, prioritizare de cerințe) și aplică aceste concepte în analiza, proiectarea și implementarea de soluții software care răspund unor cerințe date, în concordanță cu principiile de calitate și mentenabilitate.</p> <p><b>RA1</b> – Studentul/absolventul manifestă spirit de inițiativă și implicare în utilizarea și actualizarea continuă a cunoștințelor din domeniul ingineriei software (metode, procese, instrumente), adoptă o atitudine critică și responsabilă față de calitatea soluțiilor dezvoltate și selectează și utilizează în mod adecvat surse bibliografice și resurse profesionale specifice domeniului calculatoare și tehnologia informație</p>
2. Proiectarea, gestionarea ciclului de viață și integrarea sistemelor informatice utilizând tehnologii și medii de programare	<p><b>C1</b> – Studentul/absolventul identifică, descrie și sintetizează concepte și metode fundamentale privind proiectarea și gestionarea ciclului de viață al sistemelor informatice, principiile de asigurare a calității și rolul proceselor moderne (versionare, integrare continuă, testare și livrare continuă) în dezvoltarea și exploatarea aplicațiilor software.</p> <p><b>A1</b> – Studentul/absolventul elaborează specificații și proiectează aplicații software de mică dimensiune, organizează codul și artefactele într-un sistem de control al versiunilor, astfel încât să gestioneze coerent etapele ciclului de viață al unei aplicații software.</p> <p><b>RA1</b> – Studentul/absolventul arată spirit de inițiativă și acțiune în utilizarea și adaptarea practicilor moderne de inginerie software (integrări frecvente, revizuirii de cod, monitorizarea defectelor), își asumă responsabilitatea pentru calitatea și trasabilitatea livrabilelor pe parcursul ciclului de viață și selectează și utilizează în mod adecvat surse bibliografice, ghiduri și instrumente recomandate în domeniul ingineriei software</p>
<b>Competențe transversale</b>	<b>Rezultatele învățării*</b>
1. Aplicarea, în mod responsabil, a principiilor, normelor și valorilor eticii profesionale în realizarea sarcinilor profesionale și identificarea obiectivelor de realizat, a resurselor disponibile, a etapelor de lucru, a duratelor de execuție, a termenelor de realizare aferente și a riscurilor aferente.	<p><b>C1</b> – Studentul/absolventul identifică și descrie concepte fundamentale din domeniul ingineriei software și al eticii profesionale (responsabilitate față de utilizatori și organizație, confidențialitate, protecția datelor și a proprietății intelectuale)</p> <p><b>A1</b> – Studentul/absolventul identifică obiective, resurse și riscuri în cadrul unui proiect software) și formulează scenarii de lucru și de evaluare a rezultatelor, argumentând deciziile tehnice și organizatorice prin raportare la principiile eticii profesionale și la impactul asupra utilizatorilor și al organizației.</p> <p><b>RA1</b> – Studentul/absolventul are o comportare onorabilă, responsabilă și în spiritul legii în realizarea activităților specifice ingineriei software (analiză, proiectare, implementare, testare), respectă codul de etică și conduită profesională</p>
2. Identificarea rolurilor și responsabilităților într-o echipă pluridisciplinară și aplicarea de tehnici de relaționare în munca în cadrul echipei.	<p><b>C1</b> – Studentul/absolventul descrie, identifică și sintetizează concepte fundamentale din domeniul ingineriei software și al lucrului în echipe pluridisciplinare (roluri precum analist, arhitect, dezvoltator, tester, manager de proiect, stakeholderi) și explică modul în care aceste roluri colaborează în cadrul proiectelor de dezvoltare software.</p> <p><b>A1</b> – Studentul/absolventul își asumă și exercită roluri specifice într-o echipă de proiect software, aplică tehnici de comunicare și colaborare (planificare în echipă, întâlniri periodice, utilizarea de instrumente colaborative pentru cod și cerințe) și contribuie la coordonarea activităților și la integrarea rezultatelor obținute de membrii echipei.</p> <p><b>RA1</b> – Studentul/absolventul lucrează eficient ca membru al unei echipe sau în rol de coordonare la nivelul unui sub-grup, manifestă deschidere la feedback și la rezolvarea constructivă a conflictelor și descrie clar și concis, oral și în scris, rezultatele activităților proprii și ale echipei în cadrul proiectelor de inginerie software</p>

\* C – cunoștințe; A – aptitudini; RA – responsabilitate și autonomie.

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

6.1. Obiectivul general al disciplinei	➤ La sfârșitul cursului studentul va fi familiarizat, prin cunoștințele dobândite, cu problematica dezvoltării sistematice a programelor
--	--

	complexe prin aplicarea conceptelor și metodologiilor specifice ingineriei software.
6.2. Obiectivele specifice	La sfârșitul cursului, studentul va fi capabil: <ul style="list-style-type: none"> <li>➤ să înțeleagă conceptele fundamentale legate de ingineria software;</li> <li>➤ să caracterizeze principalele etape din ciclul de viață al unui produs software;</li> <li>➤ să cunoască și să utilizeze metodele și tehnicile de proiectare și dezvoltarea a aplicațiilor informatice specifice ingineriei software;</li> <li>➤ să cunoască și să utilizeze tehnicile de modelare UML.</li> </ul>

## 7. Conținuturi

7.1. Curs	Nr. ore	Metode de predare	Observații
Introducere în ingineria programelor.	2	Interactivă și convențională, centrată pe student, folosind prezentări PowerPoint, demonstrații și exemple on-line	
Etapele de dezvoltare ale programelor. Ciclul de viață al sistemelor software.	3		
Cerințe și specificații software	4		
Paradigme și metodologii de dezvoltare software. Metodologiile cascadă și spirală.	3		
Paradigme și metodologii de dezvoltare software. Metodologiile hibridă și Win-Win	3		
Paradigme și metodologii de dezvoltare software. Modelul V.	3		
Paradigme și metodologii de dezvoltare software. Programarea extremă (eXtreme Programming)	3		
Paradigme și metodologii de dezvoltare software. Dezvoltarea de software Open-Source. Modele de licențiere.	3		
Paradigme și metodologii de dezvoltare software. Reverse engineering.	3		
Proiectarea sistemelor software. Agile, DevOps, DevSecOps.	6		
Testarea sistemelor software. Determinarea cazurilor de test.	3		
Remediarea erorilor. Întreținerea aplicațiilor.	3		
Indicatori de calitate ai aplicațiilor informatice	3		
Bibliografie			
1. Pankaj Jalote, A Concise Introduction to Software Engineering – with OpenSource and GenAI, Springer, 2025			
2. Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2015;			
2. Ronald J. Leach - Introduction to Software Engineering, Second Edition, CRC Press, 2016.			
4. Cornelia Novac Ududec, Ingineria sistemelor de programe - Ingineria programării, Ediție adăugită și revizuită, Editura Alma Mater, Bacău, 2011			
7.2. Seminar / laborator	Nr. ore	Metode de predare	Observații
Identificarea și elaborarea cerințelor și specificațiilor de utilizare ale aplicațiilor informatice	4	Clasică, interactivă, centrată pe student și pe rezultatele însușirii	
Limbajul UML. Diagrama cazurilor de utilizare	4		

Limbajul UML. Diagrama de activități	4	cunoștințelor predate la curs	
Limbajul UML. Diagrama de stare	4		
Limbajul UML. Diagrama de clase	4		
Convenții pentru scrierea codului. Realizarea documentațiilor pentru aplicațiile dezvoltate	4		
Sisteme pentru controlul versiunilor software. Utilizarea GitHub.	4		
Bibliografie			
1. Pankaj Jalote, A Concise Introduction to Software Engineering – with OpenSource and GenAI, Springer, 2025			
2. Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2015;			
3. Dumitrașcu L, Gutu S. – UML 2. Studii de caz, probleme de modelare, exerciții de sinteză, Ed. Eurostampa, 2011			
4. *** - Tutorial UML - <a href="https://www.tutorialspoint.com/uml/index.htm">https://www.tutorialspoint.com/uml/index.htm</a>			
5. *** - Documentația aplicației DRAW.IO ( <a href="https://draw.io">https://draw.io</a> )			
6. Pricop, E. – Materiale pentru laborator – platforma de e-learning F-IME, UPG Ploiesti, 2025			
<b>7.3. Proiect</b>	Nr. ore	Metode de predare	Observații
<i>Nu este cazul</i>			
Bibliografie			
<i>Nu este cazul</i>			

## 8. Coroborarea conținuturilor disciplinei cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatori reprezentativi din domeniul aferent programului

Conținuturile disciplinei sunt specifice domeniului și răspund cerințelor angajatorilor, având în vedere faptul că prin studiul disciplinei se asigură o abordare inginerescă, sistematică a procesului de dezvoltare a programelor de calculator și a sistemelor informatice de complexitate medie și ridicată.

## 9. Evaluare

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	Cunoașterea noțiunilor fundamentale de analiză, proiectare, dezvoltare și evaluare a sistemelor software și a noțiunilor referitoare la metodologiile specifice aplicabile pe întregul ciclu de viață al produsului software	<p>Lucrare scrisă cu două subiecte</p> <ul style="list-style-type: none"> <li>- Subiectul 1, notat de la 1 la 10, constă în 15 întrebări cu răspuns rapid (0,6 pct. / răspuns corect + 1 pct. din oficiu).</li> <li>- Subiect 2 notat de la 1 la 10 constă în definirea și exemplificarea / aplicarea unor cunoștințe teoretice la un scenariu dat (9 pct. + 1 pct. din oficiu).</li> </ul> <p>Nota pentru lucrarea scrisă este media aritmetică a notelor pentru cele două subiecte.</p>	60%
	Evaluarea implicării active la activitatea de curs	Cuantificarea numărului de prezențe la curs, participarea activă – răspunsuri la întrebări / rezolvarea de probleme la curs, etc.	10%

9.5. Seminar/ laborator	Activitate laborator și verificări periodice	Evaluarea, atât pe parcursul semestrului a modului de realizare a lucrărilor de laborator, cât și la încheierea activității de laborator prin predarea unui proiect specific realizat în echipe de 3-4 studenți	30%
9.6. Proiect	-	-	-
9.7. Standard minim de performanță			
<ul style="list-style-type: none"> <li>➤ Studentul trebuie să demonstreze cunoașterea noțiunilor de bază privind ingineria software (cunoașterea etapelor de dezvoltare ale aplicațiilor, ciclul de viață conceptual al unui program, tipuri de cerințe);</li> <li>➤ Realizarea diagramelor UML pentru aplicația aleasă la laborator și efectuarea tuturor lucrărilor de laborator – încărcarea rezolvărilor pe platforma de e-learning;</li> <li>➤ Obținerea a cel puțin 5 puncte lucrarea scrisă și predarea diagramelor UML pentru aplicațiile de la laborator.</li> </ul>			

Data completării	Semnătura titularului de curs <i>Conf. dr. ing. Pricop Emil</i>	Semnătura titularului de seminar/laborator <i>Conf. dr. ing. Pricop Emil</i>	Semnătura titularului de proiect
19.09.2025	_____	_____	_____

Data avizării în departament	Director de departament <i>Conf. dr. ing. Pricop Emil</i>	Decan <i>Conf. dr. ing. Bădicioiu Marius</i>
26.09.2025	_____	_____