

## FIȘA DISCIPLINEI (licență)

### 1. Date despre program

Instituția de învățământ superior	Universitatea „Ștefan cel Mare” Suceava
Facultatea	Facultatea de Inginerie Electrică și Știința Calculatoarelor
Departamentul	Departamentul de Calculatoare
Domeniul de studii	Ingineria sistemelor
Ciclul de studii	Licență
Programul de studii	Automatică și informatică aplicată

### 2. Date despre disciplină

Denumirea disciplinei	<b>INGINERIA SISTEMELOR DE PROGRAME</b>				
Titularul activităților de curs	Prof.dr.ing. Cristina TURCU				
Titularul activităților de seminar	Ș.I.dr.ing. Felicia GÎZĂ				
Anul de studiu	IV	Semestrul	7	Tipul de evaluare	C
Regimul disciplinei	Categorია formativă a disciplinei DF - fundamentală, DD - în domeniu, DS - de specialitate, DC - complementară				DD
	Categorია de opționalitate a disciplinei: DI - impusă, DO - opțională, DF - facultativă				DF

### 3. Timpul total estimat (ore alocate activităților didactice)

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

II Distribuția fondului de timp pe semestru:	ore
II a) Studiul după manual, suport de curs, bibliografie și notițe	20
II b) Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren	12
II c) Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri	20
II d) Tutoriat	-
III Examinări	3
IV Alte activități:	-

Total ore studiu individual II (a+b+c+d)	52
Total ore pe semestru (Ib+II+III+IV)	125
Numărul de credite	5

### 4. Precondiții (acolo unde este cazul)

Curriculum	•
Competențe	•

### 5. Condiții (acolo unde este cazul)

Desfășurare a cursului	• PC, videoproiector și ecran, acces la Internet, G Suite, note de curs, bibliografie recomandată	
Desfășurare aplicații	Seminar	•
	Laborator/lucrări practice	• PC, videoproiector și ecran, acces la Internet, G Suite, software specializat, lucrări de laborator, bibliografie recomandată
	Proiect	•

## 6. Competențe specifice acumulate

Competențe profesionale	C2. Operarea cu concepte fundamentale din știința calculatoarelor, tehnologia informației și comunicațiilor; C5. Dezvoltarea de aplicații și implementarea algoritmilor și structurilor de conducere automată, utilizând principiile de management de proiect, medii de programare și tehnologii bazate pe microcontrolere, procesoare de semnal, automate programabile, sisteme încorporate.
Competențe transversale	•

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

Obiectivul general al disciplinei	• Însușirea de către studenți a metodelor de analiză, proiectare, implementare și testare a produselor program de calitate, cu respectarea cerințelor, cumulat cu dezvoltarea experienței de programare.
-----------------------------------	--

## 8. Conținuturi

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	4h		
5.3. UML. Diagrame comportamentale. Exemple	4h		
6. Proiectarea sistemelor software	3h		
7. Șabloane de proiectare	3h		
8. Implementarea sistemelor software			
8.1. Prezentare generală, ghiduri	2h		
8.2. Instrumente software	2h		
8.3. Reutilizare software	2h		
9. Verificare și validare. Testare software. Tipuri de testare. Instrumente pentru managementul testării	3h		
10. Calitatea produselor software	3h		
11. Securitatea sistemelor software. Securitatea în ciclul de dezvoltare a unui produs software	3h		

### Bibliografie

1. A. Cernian, A.D. Ioniță, *Noțiuni aplicative de inginerie a sistemelor de programe*, Ed. Matrixrom, București, 2009
2. Eric J. Braude, Michael E. Bernstein, *Software Engineering: Modern Approaches* 2nd Edition, 2010
3. Kenneth E. Kendall, Julie E. Kendall, *Systems Analysis and Design*, 9th Edition, 2013
4. Rod Stephens, *Beginning Software Engineering* 1st Edition, 2015
5. Max Kanat-Alexander, *Understanding Software*, 2017
6. I. Sommerville, *Software Engineering*, 10th Edition, Pearson, India, 2017
7. M. Richards and N. Ford, *Fundamentals of software architecture: An engineering approach*. Sebastopol, CA: O'Reilly Media, 2020.
8. M. Seemann, *Code that fits in your head: Heuristics for software engineering*. Boston, MA: Addison Wesley, 2021.
9. D. Farley, *Modern software engineering: Doing what works to build better software faster*. Boston, MA: Addison Wesley, 2022.

10. Materiale bibliografice specifice limbajelor de programare utilizate
Bibliografie minimală
I.Sommerville, <i>Software Engineering</i> , 10th Edition, Pearson India, 2017 O. Gherghieș, A. Apetrei, <i>Ingineria programării</i> , Curs, Iași, 2002 I. Odagescu, F. Furtuna, <i>Ingineria programării</i> , Ed. ASE, București, 2002 Fabian C., Mihalca R., Chichernea V., Goron S., Botezatu C., Iacob I., <i>Proiectarea sistemelor informatice. Metode de realizare</i> , Sylvi, Bucuresti 2001 D. Bocu – <i>Inițiere în ingineria sistemelor soft</i> , Ed. Albastră, Cluj-Napoca, 2001

Laborator/lucrări practice	Nr. ore	Metode de predare	Observații
1. Noțiuni de sănătate și securitate în muncă. Noțiuni de prim ajutor în caz de accident. Prezentarea laboratorului. Stabilire temă și echipe de lucru. Familiarizare instrumente software utile	2	lucrări practice, exerciții, studii de caz, evaluare	
2. Elaborarea specificațiilor cerințelor software	2		
3. Realizarea diagramelor UML	2		
4. Realizare Wireframes specifice interfeței utilizator. Proiectare baza de date. Stabilire arhitectura aplicație	2		
5. Elaborare plan de testare. Planificare implementare	2		
6. Considerații de implementare	2		
7. Dezvoltare module aplicație	2		
8. Implementare teste	2		
9. Populare bază de date	2		
10. Dezvoltare module aplicație	2		
11. Testare aplicație	2		
12. Rafinare implementare. Pregătire aplicație pentru beneficiar	2		
13. Testarea aplicației de către altă echipă	2		
14. Prezentare și evaluare aplicație și documentație elaborată	2		

Prezența la activitățile practice de laborator este obligatorie, conform regulamentelor USV în vigoare. Conform aceluiași regulamente, activitatea pe parcurs poate fi echivalată, la cerere, prin proiecte, pregătirea și participarea la concursuri profesionale, cu aprobarea cadrului didactic care conduce lucrările practice și cu condiția prezentării unui referat/ proiect, în care sunt descrise activitățile desfășurate și rezultatele obținute, cu evidențierea elementelor specifice tematicii disciplinei.

#### Bibliografie

1. Documentații specifice limbajelor de programare utilizate
2. I.Sommerville, <i>Software Engineering</i> , 10th Edition, Pearson India, 2017
3. A. Cernian, A.D. Ioniță, <i>Noțiuni aplicate de inginerie a sistemelor de programe</i> , Ed. Matrixrom, București, 2009
4. O. Gherghieș, A. Apetrei, <i>Ingineria programării</i> , Curs, Iași, 2002
5. I. Odagescu, F. Furtuna, <i>Ingineria programării</i> , Ed. ASE, București, 2002
6. Fabian C., Mihalca R., Chichernea V., Goron S., Botezatu C., Iacob I., <i>Proiectarea sistemelor informatice. Metode de realizare</i> , Sylvi, Bucuresti 2001
7. D. Bocu – <i>Inițiere în ingineria sistemelor soft</i> , Ed. Albastră, Cluj-Napoca, 2001
8. Booch G., Rumbaugh J., Jacobson I., <i>The Unified Modelling Language. User Guide</i> , Addison/Wesley 1999
9. D. Roman – <i>Ingineria programării obiectuale</i> , Ed. Albastră, Cluj-Napoca, 1996

#### Bibliografie minimală

1. I.Sommerville, <i>Software Engineering</i> , 10th Edition, Pearson India, 2017
2. O. Gherghieș, A. Apetrei, <i>Ingineria programării</i> , Curs, Iași, 2002
3. I. Odagescu, F. Furtuna, <i>Ingineria programării</i> , Ed. ASE, București, 2002
4. Fabian C., Mihalca R., Chichernea V., Goron S., Botezatu C., Iacob I., <i>Proiectarea sistemelor informatice. Metode de realizare</i> , Sylvi, Bucuresti 2001
5. D. Bocu – <i>Inițiere în ingineria sistemelor soft</i> , Ed. Albastră, Cluj-Napoca, 2001

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

Conținutul cursului și al laboratorului, prin problematica tratată, pune la dispoziția studentului cunoștințe conforme cu așteptările reprezentanților comunității epistemice și angajatorilor reprezentativi din domeniul calculatoarelor și tehnologiei informației. Cursul respectă recomandările IEEE și ACM pentru studiile de Software Engineering. Tematica abordată se regăsește la universități de renume, cum ar fi, University of Manchester; Truman State

University; University of Nevada; Politehnica București; Universitatea Politehnica Timișoara; Universitatea Tehnică din Cluj-Napoca; Universitatea Tehnică „Gheorghe Asachi” din Iași; Universitatea Babeș-Bolyai, Facultatea de Matematică și Informatică; Universitatea “Alexandru Ioan Cuza” din Iași, Facultatea de Informatică. Cursul se actualizează anual și în urma discuțiilor cu principalii angajatori, pentru a fi adaptat cerințelor actuale ale pieței.

#### 10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Capacitatea de utilizare adecvată a noțiunilor din domeniul ingineriei software	Evaluarea cunoștințelor teoretice și practice din tematica studiată în timpul semestrului prin test grilă Evaluarea finală a aplicației practice elaborate prin prezentare orală.	30
			20
Laborator	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	evaluare continuă a activităților desfășurate	50

##### 10.1. Standard minim de performanță evaluare la curs

- înțelegerea și cunoașterea noțiunilor fundamentale și a cel puțin 50% din cantitatea de informație vehiculată la orele de instruire
- capacitatea de a utiliza cunoștințele în rezolvarea unor probleme specifice
- elaborarea a minim 2 diagrame UML utilizând un instrument software specific
- capacitatea de a folosi corect un limbaj științific, de specialitate, adecvat disciplinei

##### 10.2. Standard minim de performanță evaluare la activitatea aplicativă

- însușirea elementelor de bază prezentate la laborator;
- capacitatea de a utiliza cunoștințele în rezolvarea unor probleme specifice;
- capacitatea de a folosi terminologia specifică domeniului și de a purta o discuție pe teme de specialitate;
- capacitatea de a lucra cu bibliotecile și framework-urile specifice domeniului, folosind unelte software dedicate.

Data completării	Semnătura titularului de curs	Semnătura titularului de aplicație
18.09.2024		

Data avizării	Semnătura responsabilului de program
20.09.2024	

Data avizării în departament	Semnătura directorului de departament
23.09.2024	

Data aprobării în consiliul facultății	Semnătura decanului
27.09.2024	