

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare și Tehnologia Informației/ Inginer
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	33.

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Inginerie software</b>				
2.2 Titularii de curs	Prof.dr.ing. Eneia Todoran – <a href="mailto:Eneia.Todoran@cs.utcluj.ro">Eneia.Todoran@cs.utcluj.ro</a>				
2.3 Titularul/ Titularii activităților de seminar/laborator/proiect	Conf.dr.info. Mitrea Paulina – <a href="mailto:Paulina.Mitrea@cs.utcluj.ro">Paulina.Mitrea@cs.utcluj.ro</a> Conf.dr.ing. Mitrea Delia - <a href="mailto:Delia.Mitrea@cs.utcluj.ro">Delia.Mitrea@cs.utcluj.ro</a>				
2.4 Anul de studiu	3	2.5 Semestrul	5	2.6 Tipul de evaluare ( E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DF – fundamentală, DD – în domeniu, DS – de specialitate, DC – complementară				DD
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

### 3. Timpul total estimat

3.1 Număr de ore pe săptămână	4	din care:	Curs	2	Seminar		Laborator	1	Proiect	1
3.2 Număr de ore pe semestru	56	din care:	Curs	28	Seminar		Laborator	14	Proiect	14
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										20
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										17
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										17
(d) Tutoriat										5
(e) Examinări										10
(f) Alte activități:										0
3.4 Total ore studiu individual (suma (3.3(a)...3.3(f)))										69
3.5 Total ore pe semestru (3.2+3.4)										125
3.6 Numărul de credite										5

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

4.1 de curriculum	Programare orientată pe obiecte, Tehnici de programare
4.2 de competențe	Competențele disciplinelor de mai sus

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

5.1. de desfășurare a cursului	Blackboard / whiteboard, internet, proiector, calculator
5.2. de desfășurare a laboratorului / proiectului	Calculator, internet, software specific

### 6. Competențele specifice acumulate

6.1 Competențe profesionale	<p><b>C3</b> - Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor</p> <p><b>C3.1</b> - Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice</p> <p><b>C3.2</b> - Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor</p> <p><b>C3.3</b> - Aplicarea tiparelor de soluții cu ajutorul uneltelor și metodelor ingineresti</p> <p><b>C3.4</b> - Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor</p>
-----------------------------	--

	<b>C3.5</b> - Dezvoltarea și implementarea de soluții informatice pentru probleme concrete
6.2 Competențe transversale	N/A

## 7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Obiectivul general al disciplinei consta in studiul si aplicarea de abordari sistematice, disciplinate si cuantificabile in dezvoltarea sistemelor software
7.2 Obiectivele specifice	<p>Pentru atingerea obiectivului general se urmaresc obiective specifice:</p> <ul style="list-style-type: none"> <li>• Studiul si aplicarea proceselor de dezvoltare software</li> <li>• Intelegerea activitatilor specifice ingineriei software</li> <li>• Cunoasterea metodelor ingineriei software</li> <li>• Cunoasterea unor instrumente specifice ce asista inginerul software in procesul de specificare, proiectare si validare</li> <li>• Cunoasterea unor metode de modelare si analiza performanta software</li> <li>• Aplicarea proceselor, metodelor si instrumentelor studiate in proiecte software de dimensiuni mici si medii</li> </ul>

## 8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Introducere și privire de ansamblu asupra domeniului	2	Prezentari PowerPoint, exemple, intrebari, discutii (pentru interactiune online se utilizeaza Teams, Discord)	
Paradigme de dezvoltare software; paradigme de bază: ‘cascada’, prototipizare, componente reutilizabile, metode formale;	2		
Paradigme evolutive: dezvoltare incrementală, model spirală, inginerie concurentă	2		
Procese moderne: procesul unificat, metode flexibile și programare extrema; activități de bază (specificare, dezvoltare, validare, evoluție): concepte, principii, procese	2		
Dezvoltare cerințe: analiza de domeniu, tipuri de cerințe, tehnici de obținere a cerințelor, captarea cerințelor sub forma de cazuri de utilizare	2		
Analiza si specificare formala; principiile model checking; analiza performanta software	2		
Instrumente pentru analiza si specificare formala; PRISM probabilistic model checker	2		
Modelare cu clase: diagrame UML de clase, procesul de construire a diagramelor de clase, semantica diagramelor UML de clase, implementarea diagramelor de clase în Java	2		
Utilizare șabloane de proiectare (design patterns: Adapter, Composite, Facade, Observer, etc)	2		
Modelare interacțiuni și comportament: diagrame UML de interacțiuni (secvențiere și colaborare), stare și activitate	2		
Proiectare si arhitectura software: principii (creșterea gradului de coeziune, reducerea gradului de cuplare, etc.), șabloane arhitecturale (Multi-Layer, Pipe-and-Filter, etc.)	2		
Testare software: tehnici de testare (partiționare în clase de echivalenta, testarea cailor program, etc.) și strategii de integrare (top-down, bottom-up, bazata pe scenarii de utilizare)	2		
Dezvoltare ghidata de cazurile de utilizare: specificare prin cazuri de utilizare, analiza, proiectare și implementare pentru realizarea cazurilor de utilizare, testarea cazurilor de utilizare	2		
Specificații program: pre- și post- condiții, inducție (well founded induction), prototipizare declarativă	2		
<p><b>Bibliografie</b> (<i>bibliografia minimală a disciplinei conținând cel puțin o lucrare bibliografică de referință a disciplinei, care există la dispoziția studenților într-un număr de exemplare corespunzător</i>)</p> <p>Bibliography</p> <p>1. I. Sommerville, <i>Software Engineering</i> (6<sup>th</sup>, 7<sup>th</sup>, 8<sup>th</sup>, 9<sup>th</sup>, 10<sup>th</sup> editions), Addison Wesley (2001, 2004, 2006, 2010, 2015).</p>			

2. T. Lethbridge, R. Laganier, <i>Object-Oriented Software Engineering: Practical Software Development using UML and Java</i> (2 <sup>nd</sup> edition), McGraw-Hill, 2005. <a href="http://www.lloseng.com">http://www.lloseng.com</a> .			
3. A. Fox, D. Patterson, <i>Engineering Software as a Service: An Agile Approach Using Cloud Computing</i> , Strawberry Canon, 2016.			
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, <i>Design Patterns: Elements of Reusable Object-Oriented Software</i> , Addison-Wesley, 1994.			
5. E.M. Clarke, T.A. Henzinger, H. Veith, R. Bloem, editors, <i>Handbook of Model Checking</i> , Springer, 2018.			
6. M. Odersky, L. Spoon, B. Venners, <i>Programming in Scala</i> , (3 <sup>rd</sup> , 4 <sup>th</sup> editions), Artima (2016, 2020).			
7. E.N. Todoran, <i>Inginerie software: studii in prototipizare si specificare formala</i> , Mediamira, Cluj-Napoca, 2006.			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații
OCSF – framework client-server pentru dezvoltare prin reutilizare	2		
Simple Chat – sistem instant messaging bazat pe OCSF (1)	2		
Simple Chat – sistem instant messaging bazat pe OCSF (2)	2		
Utilizare instrumente CASE de modelare software: diagrame UML de clase, cazuri de utilizare, interacțiune, stare, desfasurare	2		
Utilizare instrumente CASE de modelare si analiza performanta: PRISM model checker	2		
Utilizare Design Patterns	2		
Proiectare cazuri de test, utilizare JUnit	2		
<b>Activitatea de dezvoltare a proiectului</b> implica utilizarea paradigmelor si instrumentelor software prezentate la curs. Este incurajata activitatea in echipa de dezvoltatori software. Studentii utilizeaza paradigme si metode de dezvoltare software prezentate la curs, de exemplu, urmand modelul SaaS (Software as a Service). Exista trei iteratii cu termene predefinite. In cazul abordarii traditionale de tip 'cascada' (modelul linear) termenele corespund activitatilor de specificare a cerintelor, design detaliat si livrare finala. Proiectul se preda in saptamana 13.	14		
<b>Bibliografie</b> (bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător)			
1. T. Lethbridge, R. Laganier, <i>Object-Oriented Software Engineering: Practical Software Development using UML and Java</i> (2 <sup>nd</sup> edition), McGraw-Hill, 2005. <a href="http://www.lloseng.com">http://www.lloseng.com</a> .			
2. E. Gamma, R. Helm, R. Johnson, J. Vlissides, <i>Design Patterns: Elements of Reusable Object-Oriented Software</i> , Addison-Wesley, 1994.			
3. PRISM manual, 2019. <a href="http://www.prismmodelchecker.org/manual/">http://www.prismmodelchecker.org/manual/</a>			

\*Se vor preciza, după caz: tematica seminariilor, lucrările de laborator, tematica și etapele proiectului.

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

Ingineria software este o disciplina de baza in domeniul Calculatoare si Tehnologia Informatiei. In cadrul cursului, a lucrarilor practice si a orelor de proiect studentii fac cunostinta cu procese, metode si instrumente specifice, si invata sa aplice abordari sistematice si cuantificabile in dezvoltarea sistemelor software. Continutul disciplinei a fost elaborat in baza interactiunii cu specialisti in domeniul Ingineriei Software din Romania, Europa (UK, Greece), SUA si Canada si a fost evaluat de agentii guvernamentale romanesti (CNEAA si ARACIS).

### 10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilitati de rezolvare probleme	Examen final (pentru evaluare online se utilizeaza Teams, Moodle, Discord, Email)	75%
Seminar			
Laborator	Abilitati de proiectare si validare in cadrul unui proiect software	Test laborator, evaluare proiect (pentru evaluare online se utilizeaza Teams, Moodle, Discord, Email)	5% 20%
Proiect			
Standard minim de performanță:			

Realizarea unui proiect software de dimensiuni medii utilizand cunostintele predate la cursul de Inginerie Software.  
Calcul nota disciplina: 5% laborator + 20% proiect + 75% examen final  
Conditii de participare la examenul final: Laborator  $\geq$  5, Proiect  $\geq$  5  
Conditii de promovare: Nota  $\geq$  5

<b>Data completării:</b>	<b>Titulari</b>	<b>Titlu Prenume NUME</b>	<b>Semnătura</b>
15.09.2020	Curs	Prof.dr.ing. Enea Todoran	
	Aplicații	Conf.dr.info. Mitrea Paulina Conf.dr.ing. Mitrea Delia	

Data avizării în Consiliul Departamentului Calculatoare	Director Departament Prof.dr.ing. Rodica Potolea
Data aprobării în Consiliul Facultății de Automatică și Calculatoare	Decan Prof.dr.ing. Liviu Miclea