

## 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 Ciclu de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare română/ Inginer
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	41.00

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Proiectare software</b>				
2.2 Titularii de curs	Prof. dr. ing. Mihaela Dîșoreanu - <a href="mailto:mihaela.dinsoreanu@cs.utcluj.ro">mihaela.dinsoreanu@cs.utcluj.ro</a>				
2.3 Titularul / Titularii activităților de Seminar / laborator / proiect	Șl. dr. info. Anca Iordan - <a href="mailto:anca.iordan@cs.utcluj.ro">anca.iordan@cs.utcluj.ro</a>				
2.4 Anul de studiu	III	2.5 Semestrul	6	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				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DI

### 3. Timpul total estimat

3.1 Număr de ore pe săptămână	5	din care:	Curs	2	Seminar		Laborator	2	Proiect	1
3.2 Număr de ore pe semestru	70	din care:	Curs	28	Seminar		Laborator	28	Proiect	14
3.3 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										10
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										5
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										6
(d) Tutoriat										4
(e) Examinări										5
(f) Alte activități:										
3.4 Total ore studiu individual (suma (3.3(a)...3.3(f)))							30			
3.5 Total ore pe semestru (3.2+3.4)							100			
3.6 Numărul de credite							4			

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

4.1 de curriculum	Tehnici de programare, Inginerie software
4.2 de competențe	Metodologii de proiectare, Structuri de date, Șabloane de proiectare de bază

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

5.1. de desfășurare a cursului	Tablă, proiector, calculator conectat la internet, Teams, Moodle.
5.2. de desfășurare a laboratorului / proiectului	16 Calculatoare, software specific, proiector, Teams, GitHub Prezența la laborator și proiect este obligatorie

### 6. Competențele specifice acumulate

6.1 Competențe profesionale	<b>C3</b> - Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor <ul style="list-style-type: none"> <li>• <b>C3.1</b> - Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice</li> <li>• <b>C3.2</b> - Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea</li> </ul>
-----------------------------	---

	rezultatelor lor <ul style="list-style-type: none"> <li>• <b>C3.3</b> - Aplicarea tiparelor de soluții cu ajutorul uneltelor și metodelor ingineresti</li> <li>• <b>C3.4</b> - Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor.</li> <li>• <b>C3.5</b> - Dezvoltarea și implementarea de soluții informatice pentru probleme concrete.</li> </ul>
6.2 Competențe transversale	N/A

## 7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Obiectivul general al disciplinei este prezentarea si analiza solutiilor arhitecturale pe diferite nivele de abstractizare
7.2 Obiectivele specifice	<ul style="list-style-type: none"> <li>• Identificați cele mai relevante cerințe funcționale și nefuncționale ale unui sistem software și documentați-le</li> <li>• Înțelegerea principiilor de proiectare a clasei și a pachetelor</li> <li>• Analizați arhitecturile software în raport cu principiile de proiectare cunoscute</li> <li>• Recunoașteți stilurile arhitecturale fundamentale ale software-ului și modelele de design</li> <li>• Proiectați arhitecturi software adecvate pe baza cerințelor date</li> </ul>

## 8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Introducere. Principii de proiectare a claselor (SOLID)	2	Prezentări, PowerPoint, Quizuri, teme si discutii, materiale de curs Moodle	
Principii de proiectare a claselor (GRASP) si pachetelor	2		
Sabloane arhitecturale (Layers, Event-driven, MVC)	2		
Domain-driven design	2		
Service-oriented design	2		
Examen partial/Demonstratie de cod	2		
Sabloane aplicate in arhitecturi Enterprise (Accesul la resurse)	2		
Sabloane aplicate in arhitecturi Enterprise (Prezentare)	2		
Sabloane aplicate in arhitecturi Enterprise (Concurenta)	2		
Aplicarea sabloanelor de proiectare (creationale)	2		
Aplicarea sabloanelor de proiectare (structurale)	2		
Aplicarea sabloanelor de proiectare (comportamentale)	2		
Atribute de calitate si Metrici ale calitatii proiectelor software	2		
Recapitulare si pregatire pentru examen	2		
Bibliografie ( <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> ) <ol style="list-style-type: none"> <li>1. Juval Lowy, <i>Righting software</i>, O'Reilly, 2020</li> <li>2. Mark Richards, <i>Software Architecture Patterns</i>, O'Reilly, 2015</li> <li>3. Vaughn Vernon, <i>Domain Driven Design Distilled</i>, Addison Wesley, 2016</li> <li>4. Ian Gorton, <i>Essential Software Architecture</i>, Springer, second ed. 2011.</li> <li>5. Taylor, R., Medvidovic, N., Dashofy, E., <i>Software Architecture: Foundations, Theory, and Practice</i>, 2010, Wiley.</li> <li>6. Len Bass, Paul Clements, Rick Kazman, <i>Software Architecture in Practice</i>, 3rd edition, 2013.</li> <li>7. Buschmann, Frank, Regine Meunier, Hans Rohnert, Peter Sornmerlad, and Michael Stal. 2001. <i>Pattern-oriented system architecture, volume 1: A system of patterns</i>. Hoboken, NJ: John Wiley &amp; Sons.</li> <li>8. Fowler Martin, <i>Patterns of Enterprise Application Architecture</i>, Addison-Wesley Professional, 2002</li> <li>9. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. <i>Design Patterns</i>. AddisonWesley, 1995.</li> <li>10. Craig Larman, <i>Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development</i> (3rd Edition), Prentice Hall, 2004, ISBN: 0131489062 Materialele de curs sunt publicate pe moodle.cs.utcluj.ro</li> </ol>			
8.2 Aplicații (seminar/laborator/proiect)*	Nr.ore	Metode de predare	Observații

Exercitii recapitulative (OOP, UML, tehnici de testare). Principiile SOLID	3	Implementarea temelor pe sistemele din laborator/GitHub, prezentari si demonstratii la tabla	
Conexiunea la baze de date și operații specifice	3		
GRASP. Șablonul arhitectural MVP	3		
Șabloanele arhitecturale MVC și MVVM	3		
Domain driven design - Entities, aggregates, repositories	3		
Arhitecturi orientate pe servicii	3		
Conectarea BL la structura bazei de date	3		
eXtensible Markup Language si JSON	3		
Elemente de dezvoltare front-end	3		
Sabloane de proiectare creationale	3		
Sabloane de proiectare structurale	3		
Sabloane de proiectare comportamentale	3		
Recuperari	3		
Recapitulare si pregatire examem	3		
Bibliografie ( <i>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</i> )			
<ul style="list-style-type: none"> <li>• Indrumator de laborator</li> <li>• Tutorialul de Java – docs.oracle.com</li> <li>• Tutorial de C# - msdn.microsoft.com</li> </ul>			

\*Se vor preciza, după caz: tematica seminarilor, 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

Disciplina este o disciplină de domeniu în Calculatoare și Tehnologia Informației, conținutul ei fiind fundamental în proiectarea soluțiilor software. Conținutul disciplinei conține atât soluțiile arhitecturale fundamentale cât și soluții moderne care adresează complexitatea actuală a sistemelor software. Conținutul este compatibil cu disciplinele similare predate la universități de prestigiu din țară și străinătate. În elaborarea conținutului au fost consultate companii importante din România și a fost evaluat de agenții guvernamentale românești (CNEAA și ARACIS).

### 10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
Curs	Abilități de înțelegere a cerințelor, analiza a soluțiilor alternative și proiectare a soluției potrivite, prezența, activitate (evaluare_curs)	Examen scris, rezolvare Quizuri Moodle pe parcurs	p = 60%
Seminar	-	-	-
Laborator	Abilitati analiza a cerințelor și soluțiilor alternative, proiectare și implementare soluție, prezența, activitate (evaluare_lab+proiect)	Teme de laborator, proiect GitHub	1-p
Proiect			

Standard minim de performanță:

Calcul nota disciplina:  $p \cdot \text{evaluare\_curs} + (1-p) \cdot \text{evaluare\_lab+proiect}$

Condiții de participare la examenul final: Nota Laborator  $\geq 5$  SI Nota Proiect  $\geq 5$  Condiții

de promovare: nota disciplina  $\geq 5$ , evaluare\_curs  $\geq 5$

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
22.05.2024	Curs	Prof.dr.ing. Mihaela Dînsoreanu	
	Aplicații	ȘI.dr.info. Anca-Elena Iordan	

Data avizării în Consiliul Departamentului Calculatoare  
20.02.2024

Director Departament,  
Prof.dr.ing. Rodica Potolea

Data aprobării în Consiliul Facultății de Automatică și Calculatoare  
22.02.2024

Decan,  
Prof.dr.ing. Mihaela Dînșoreanu