

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 română / Inginer
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Proiectare software				Codul disciplinei	41
2.2 Titularul de curs	Prof. dr. ing. Mihaela Dîșoreanu - mihaela.dinsoreanu@cs.utcluj.ro					
2.3 Titularul activităților de seminar / laborator / proiect	Șl. dr. info. Anca Iordan - anca.iordan@cs.utcluj.ro As. drd. ing. Bogdan Bindea - bogdan.bindea@cs.utcluj.ro					
2.4 Anul de studiu	III	2.5 Semestrul	5	2.6 Tipul de evaluare	E	
2.7 Regimul disciplinei	Categorია formativă: <i>DF – fundamentală, DS - de specialitate, DC - complementară</i>					DS
	Opționalitate: <i>DOB - obligatorie, DOP - opțională, DFA - facultativă</i>					DOB

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	C3 - Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor C3.1 - Identificarea unor clase de probleme și metode de rezolvare caracteristice sistemelor informatice C3.2 - Utilizarea de cunoștințe interdisciplinare, a tiparelor de soluții și a uneltelor, efectuarea de experimente și interpretarea rezultatelor lor C3.3 - Aplicarea tiparelor de soluții cu ajutorul uneltelor și
-----------------------------	---

	metodelor ingineresti • C3.4 - Evaluarea comparativă, inclusiv experimentală, a alternativelor de rezolvare, pentru optimizarea performanțelor. • C3.5 - Dezvoltarea și implementarea de soluții informatice pentru probleme concrete.
6.2 Competențe transversale	N/A

7. Rezultatele așteptate ale învățării

Cunoștințe	Describe, identifică și sumarizează concepte și metode de baza privitoare la principii de proiectare a arhitecturilor software, principalele stiluri arhitecturale si sabloane de proiectare fundamentale, și anume: • Principii: de proiectare a claselor si pachetelor • Stiluri arhitecturale: Layered, Pipeline, bazate pe evenimente, orientate pe servicii, microservicii • Sabloane de proiectare: creationale, structurale si comportamentale
Aptitudini	• Analizeaza cerinte si identifica cerintele functionale si caracteristici arhitecturale • Analizeaza alternative de solutii arhitecturale pentru o problema specifica, alege o solutie arhitecturala si justifica alegerea aratand punctele tari si slabe ale solutiei in raport cu caracteristicile arhitecturale de interes • Proiecteaza solutii specifice integrand stiluri arhitecturale si sabloane de proiectare adecvate • Documenteaza proiectele elaborate in vederea implementarii
Responsabilitate și autonomie:	Are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.

8. Obiectivele disciplinei

8.1 Obiectivul general al disciplinei	Obiectivul general al disciplinei este prezentarea si analiza solutiilor arhitecturale pe diferite nivele de abstractizare
8.2 Obiectivele specifice	• Identificarea celor mai relevante cerințe funcționale și caracteristici arhitecturale ale unui sistem software și documentarea lor • Înțelegerea principiilor de proiectare a clasei și a pachetelor • Analiza arhitecturilor software în raport cu principiile de proiectare cunoscute • Înțelegerea stilurilor arhitecturale fundamentale ale software-ului și a modelelor de proiectare • Proiectarea arhitecturilor software adecvate pe baza cerințelor date

9. Conținuturi

9.1 Curs	Nr.ore	Metode de predare	Observații
Introducere. Principii de proiectare a claselor (SOLID)	2	Prezentări, Power-Point, Quizuri, teme si discutii, materiale de curs Moodle.	
Principii de proiectare a claselor (GRASP) si pachetelor	2		
Caracteristici arhitecturale	2		
Stiluri arhitecturale (Layers, Pipeline, MVC)	2		
Stiluri arhitecturale (Microkernel, Service based, DDD)	2		
Stiluri arhitecturale (Event-driven (Broker, Mediator))	2		
Stiluri arhitecturale (Orchestration driven (SOAP, REST))	2		
Stiluri arhitecturale (Choreography driven (Microservices))	2		
Stiluri arhitecturale (Volatility driven)	2		
Accesul la date	2		
Aplicarea sabloanelor de proiectare (creationale)	2		
Aplicarea sabloanelor de proiectare (structurale)	2		
Aplicarea sabloanelor de proiectare (comportamentale)	2		
Recapitulare si pregatire pentru examen	2		

Bibliografie (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)

1. **Mark Richards, Neal Ford, Fundamentals of Software Architecture, 2020**
2. Juval Lowy, Righting software, O'Reilly, 2020
3. Vaughn Vernon, Domain Driven Design Distilled, Addison Wesley, 2016
4. Fowler Martin, *Patterns of Enterprise Application Architecture*, Addison-Wesley Professional, 2002
5. E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns. AddisonWesley, 1995.
6. Craig Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* (3rd Edition), Prentice Hall, 2004, ISBN: 0131489062

Materialele de curs sunt publicate pe moodle.cs.utcluj.ro

9.2 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.	
Principii de proiectare a claselor și pachetelor	3		
Exercitii de analiza a cerintelor: identificare caracteristici arhitecturale	3		
Șabloanele arhitecturale MVC, MVP, MVVM	3		
Microkernel si Domain driven design	3		
Broker si Mediator	3		
Orchestrare (SOAP si REST)	3		
Microservicii	3		
Metoda iDesign	3		
Conectarea logicii de business la structura de date	3		
Sabloane de proiectare creationale	3		
Sabloane de proiectare structurale	3		
Sabloane de proiectare comportamentale	3		
Recuperari	3		

Bibliografie (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)

- Indrumator de laborator
- Tutorialul de Java – docs.oracle.com
- Tutorial de C# - msdn.microsoft.com

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

10. 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).

11. 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, prezenta, activitate (evaluare_curs)	Examen scris, rezolvare Quizuri Moodle pe parcurs	p = 60% din care Quiz = 10% Examen = 50%
Seminar	-	-	-
Laborator	Abilitati analiza a cerintelor si solutiilor alternative, proiectare si implementare solutie, prezenta, activitate (evaluare_lab+proiect)	Teme de laborator, proiect GitHub	1-p
Proiect			

Standard minim de performanță:

Calcul nota disciplina: $p * \text{evaluare_curs} + (1-p) * \text{evaluare_lab+proiect}$

Conditii de participare la examenul final: Nota Laborator ≥ 5 SI Nota Proiect ≥ 5 Conditii

de promovare: nota disciplina ≥ 5 , Examen ≥ 5

Data completării: 15.04.2026	Titulari	Titlu, prenume / NUME	Semnătura
	Curs	Prof.dr.ing. Mihaela DÎNȘOREANU	
	Aplicații	Șl.dr. Anca-Elena IORDAN	
		As.drd.ing. Bogdan BINDEA	

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. Vlad Mureșan