

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Facultatea de Inginerie
1.3 Departamentul	Inginerie Electrică, Electronică și Calculatoare
1.4 Domeniul de studii	INGINERIE ELECTRICĂ
1.5 Ciclul de studii	Licență - dual
1.6 Programul de studii / Calificarea	INFORMATICĂ APLICATĂ ÎN INGINERIE ELECTRICĂ
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	53.10

2. Date despre disciplină

2.1 Denumirea disciplinei	Aplicații în informatică				
2.2 Titularul de curs	Prof. dr. ing. Matei Oliviu - Oliviu.MATEI@ieec.utcluj.ro				
2.3 Titularul activităților de seminar / laborator / proiect	Dr. ing. Erdei Rudolf				
2.4 Anul de studiu	4	2.5 Semestrul	1	2.6 Tipul de evaluare	C
2.7 Regimul disciplinei	Categoria formativă				DS
	Opționalitate				DO

3. Timpul total estimat

3.1 Număr de ore pe săptămână	5	din care:	IIS	3.2 Curs	2	3.3	0	3.3	0	3.3	0
			OE		0	Seminar	0	Laborator	1	Proiect	2
3.4 Număr de ore pe semestru	70	din care:	IIS	3.5 Curs	28	3.6	0	3.6	0	3.6	0
			OE		0	Seminar	0	Laborator	14	Proiect	28
3.7 Distribuția fondului de timp (ore pe semestru) pentru:									IIS		OE
(a) Studiul după manual, suport de curs, bibliografie și notițe									10		20
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren									8		18
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri									2		18
(d) Tutoriat											
(e) Examinări									2		2
(f) Alte activități:											
Total ore studiu individual (suma (3.7(a)...3.7(f)))						80		22		58	
3.8 Total ore pe semestru (3.4+3.7)						150		50		100	
3.9 Numărul de credite						6		2		4	

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Sala trebuie să fie dotată cu tablă și videoproiector.
4.2 de competente	

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	
5.2. de desfășurare a seminarului / laboratorului / proiectului	

6.a. Competențele specifice acumulate

Competențe profesionale	AC9. Dezvoltă software cu sursă deschisă. AC16. Gândește în mod abstract.
Competențe transversale	AC06. Colaborează în echipe și rețele

6.b. Rezultatele așteptate ale învățării

Cunoștințe	Studentul/absolventul descrie, identifică și sumarizează concepte fundamentale de automatică, metodele de modelare, identificare, simulare și analiză a proceselor și a tehnicilor de proiectare asistată de calculator a sistemelor automate clasice și inteligente și modul lor de aplicare în probleme concrete.
Aptitudini	Studentul/absolventul interpretează și explică problemele de automatizare a unor tipuri de procese folosind principiile de bază din teoria sistemelor, ingineria reglării automate, modelare și simulare, tehnicile de proiectare asistată de calculator și metodele de analiză specifice ingineriei sistemelor. Studentul/absolventul recunoaște și implementează arhitecturi complexe pentru sisteme autonome inteligente. Studentul/absolventul specifică cerințe, elaborează scenarii de simulare, propune soluții de rezolvare a unor probleme de control, analizează și evaluează performanțele sistemelor automate.
Responsabilitate și autonomie:	Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.

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

7.1 Obiectivul general al disciplinei	- Cunoașterea de către studenți a principalelor componente ale arhitecturilor de sisteme de programe complexe. - Dobândirea de către studenți a capacității de a utiliza modele abstracte de reprezentare a unui domeniu, atât pentru descrierea structurală (modelarea structurilor de date), cât și pentru descrierea funcțională (modelarea proceselor) a domeniului de aplicație considerat. - Cunoașterea diverselor metodologii de dezvoltare, respectiv capacitatea de a alege metodologia de dezvoltare potrivită fiecărui proiect în parte.
7.2 Obiectivele specifice	Evaluarea comparativă, inclusiv experimentală, a alternativelor de dezvoltare, pentru optimizarea performanțelor sistemelor informatice complexe;

	Lucrările practice și proiectul se bazează pe rezolvarea de aplicații, pe sisteme test cât și pe utilizarea pachetului de programe Enterprise Architect, cu ajutorul căruia se dezvoltă aplicații specifice.
--	--

8. Conținuturi

8.1 Curs	Nr. ore	Metode de predare	Observații	
1. Introducere Istoric, evoluție, terminologie, utilizatori, limbaje. Arhitecturi si metodologii de dezvoltare – discutii generale.	2	PBL – Problem Based Learning CBL – Case Based Learning		
2. Arhitecturi Niveluri de abstractizare: nivelul domeniului de aplicatie; nivelul conceptual (independent de tehnologie), nivelul logic (al claselor de tehnologii), nivelul fizic (specific tehnologiilor de implementare). Reprezentari specifice diferitelor niveluri arhitecturale pentru: date, procese, locatii, utilizatori, evenimente, reguli de functionare.	6			
3. Dezvoltarea sistemelor de programe bazata pe model Arhitecturi bazate pe model (MDA): Principiul independentei intre nivelurile arhitecturale. Modelarea cerintelor (domain models). Proiectarea sistemelor independent de platforma (platform independent model). Proiectul specific unei platforme (platform specific model). Limbaje de modelare consacrate: unified modeling language, entity-relationship model, business process modelling notation.	10			
4. Metodologii de dezvoltare Metodologii, evoluție, avantaje si dezavantaje: metodologii de dezvoltare in cascada, in spirala, de tip RAD (rapid application development), de tip AGILE. Metrici de evaluare.	6			
5. Implementarea sistemelor de programe Alegerea tehnologiilor de implementare: performanta, accesibilitate, usurinta in utilizare, compatibilitate intre platforme. Testare si dare in productie.	4			
Bibliografie 1. Marius Muji - Arhitectura sistemelor informationale - notite de curs, 2020 2. Joseph S. Valacich, Joey F. George, and Jeffrey A. Hoffer, Essentials of Systems Analysis and Design: Pearson Education Limited, 2015 3. John W. Satzinger, Robert B. Jackson, Stephen D. Burd, Systems Analysis and Design in a Changing World, Course Technology, 7th edition, 2015 4. Pressman - Software Engineering: A Practitioner’s Approach : McGraw-Hill Education; 8 edition, 2014. 5. Object Management Group – MDA Specifications - http://www.omg.org/mda/specs.htm . 6. David C. Hay, Data Model Patterns: A Metadata Map.: Morgan Kaufmann, 2006. 7. Bruegge, Dutoit, Object-Oriented Software Engineering using UML, Patterns, and Java: - Pearson; 3 edition, 2009				
8.2	Nr. ore IIS	Nr. ore OE	Metode de predare	Observații
Laborator			Aplicația, experimentul	
1. Clasificarea arhitecturilor software și analiza unui sistem real		2		
2. Modelarea cerințelor – Use Case și Domain Model (UML)		2		
3. Platform Independent Model (PIM)		2		

4. Platform Specific Model (PSM)		2		
5. Metodologii de dezvoltare software – simulare SCRUM Agile		2		
6. Modelare proces cu BPMN (Business Process Modeling Notation)		2		
7. Compararea tehnologiilor de implementare și testare		2		
Proiect				
1. Aplicație de rezervare online pentru evenimente sau săli de clasă		4		
2. Sistem de management documente (DMS)		4		
3. Platformă de feedback pentru cursuri universitare		4		
4. Aplicație de tip HelpDesk cu suport pentru ticketing		4		
5. Sistem de raportare abateri etice / sesizări interne		4		
6. Aplicație web pentru gestiune de resurse educaționale		4		
7. Sistem de urmărire activități într-un proiect (Project Tracker)		4		
Bibliografie 1. Marius Muji - Arhitectura sistemelor informationale - notite de curs, 2020 2. Joseph S. Valacich, Joey F. George, and Jeffrey A. Hoffer, Essentials of Systems Analysis and Design: Pearson Education Limited, 2015 3. John W. Satzinger, Robert B. Jackson, Stephen D. Burd, Systems Analysis and Design in a Changing World, Course Technology, 7th edition, 2015 4. Pressman - Software Engineering: A Practitioner's Approach : McGraw-Hill Education; 8 edition, 2014. 5. Object Management Group – MDA Specifications - http://www.omg.org/mda/specs.htm . 6. David C. Hay, Data Model Patterns: A Metadata Map.: Morgan Kaufmann, 2006. 7. Bruegge, Dutoit, Object-Oriented Software Engineering using UML, Patterns, and Java: - Pearson; 3 edition, 2009				

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

Conținutul disciplinei este coroborat cu necesitățile angajatorilor din domeniile ingineriei sistemelor automate, respectiv al automatizării și informaticii aplicate. În vederea stabilirii conținuturilor, alegerii metodelor de predare/învățare titularii disciplinei organizează periodic întâlniri cu reprezentanți ai mediului economic din domeniu pentru identificarea nevoilor și angajatorilor din și coordonarea cu alte programe similare din cadrul altor instituții de învățământ superior.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoașterea noțiunilor și conceptelor fundamentale. Capacitatea de a interconecta diferite componente.	Teste grila	50%
10.5 Seminar / Laborator /Proiect	Realizarea activităților de pregătire pe parcursul semestrului	Evaluare pe parcurs	50%
10.6 Standard minim de performanță • Cunoașterea noțiunilor de bază prezentate la curs și obținerea minim a notei 5 la evaluarea finală. Prezenta la laboratoare presupune admiterea la examen. • Realizarea de lucrări sub coordonare, pentru rezolvarea unor probleme specifice domeniului, cu evaluarea corectă a volumului de lucru, a resurselor disponibile, a timpului necesar de finalizare și a			

riscurilor, în condiții de aplicare a normelor deontologice si de etica profesionala în domeniu, precum si de securitate si sănătate în munca

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
12.02.2025	Curs	<i>Prof. dr. ing. Matei Oliviu</i>	
	Aplicații	<i>Dr. ing. Erdei Rudolf</i>	

Data avizării în Consiliul Departamentului	Director Departament
17.03.2025	Conf.dr.ing. Claudiu LUNG
Data aprobării în Consiliul Facultății	Decan
09.04.2025	Conf.dr.ing. Olivian CHIVER