

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	Automatică
1.4 Domeniul de studii	Ingineria sistemelor
1.5 Ciclul de studii	Master
1.6 Programul de studii / Calificarea	Informatica Aplicata
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Structuri software pentru aplicații de timp real			Codul disciplinei	13.10
2.2 Titularul de curs	Conf. Dr. Ing. Mihai HULEA				
2.3 Titularul activităților de seminar / laborator / proiect	Conf. Dr. Ing. Mihai HULEA				
2.4 Anul de studiu	1	2.5 Semestrul	1	2.6 Tipul de evaluare	E
2.7 Regimul disciplinei	Categorica formativă				DA
	Opționalitate				DO

3. Timpul total estimate

3.1 Număr de ore pe săptămână	3	din care:	3.2 Curs	2	3.3 Seminar		3.3 Laborator	1	3.3 Proiect	
3.4 Număr de ore pe semestru	42	din care:	3.5 Curs	28	3.6 Seminar		3.6 Laborator	14	3.6 Proiect	
3.7 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										20
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										30
(d) Tutoriat										10
(e) Examinări										3
(f) Alte activități:										
3.8 Total ore studiu individual (suma (3.7(a)...3.7(f)))				83						
3.9 Total ore pe semestru (3.4+3.8)				42						
3.10 Numărul de credite				5						

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	NA
4.2 de competențe	NA

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Prezentarea unei aplicații concepute cu o metoda studiată la curs
5.2. de desfășurare a seminarului / laboratorului / proiectului	Prezența la laborator este obligatorie

6. Competențele specifice acumulate

Competențe profesionale	- Dezvoltă software cu sursă deschisă - Sintetizează informații - Găsește soluții pentru probleme - Asigura conformitate cu specificațiile
-------------------------	---

Competențe transversale	Dă dovadă de inițiativă
-------------------------	---

7. Rezultatele așteptate ale învățării

Cunoștințe	Studentul va cunoaște: - modelele și tehnicile de analiză a componentelor software - tehnici și principii de analiză, proiectare și implementare a aplicațiilor informatice bazate pe echipamente programabile și sisteme încorporate - metodele de simulare, modelare, evaluare și optimizare a performanțelor aplicațiilor software
Abilități	Studentul va fi capabil: - să proiecteze, dezvolte și testeze aplicații informatice și sisteme complexe de control utilizând metode și tehnologii moderne - identifice și aplice soluții inovative pentru optimizarea aplicațiilor informatice complexe - să realizeze prototipuri software și/sau hardware funcționale pentru demonstrarea rezultatelor cercetării
Responsabilitate și autonomie:	Studentul: - Va fi responsabil pentru gestionarea corectă a resurselor de proiect, planificarea activităților și evaluarea riscurilor asociate - Va gestiona autonom proiecte de inginerie software, industrială și sisteme încorporate

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

8.1 Obiectivul general al disciplinei	Conceperea și verificarea aplicații de timp-real de nivel ridicat transmiterea de cunoștințe de programare concurentă și programare de timp real, familiarizarea cu mediul de programare Java și instruirea în vederea realizării de aplicații de timp real. Introducere în calculul cuantic și aplicații cuantice
8.2 Obiectivele specifice	În urma parcurgerii cursului, studenții trebuie să cunoască următoarele: - Conceperea aplicațiilor mixte: classic/cuantic; - Implementarea concurentă a aplicațiilor distribuite de timp-real - Utilizarea unor metode și framework-uri pentru dezvoltarea aplicațiilor complexe

9. Conținuturi

9.1 Curs	Nr. ore	Metode de predare	Observații
Concepte de baza OOP	2	Expunere, prezentare, discuții laptop, proiector, dezbateri, curs interactiv / în caz de forță majoră, on-line platforma Teams	NA
Stocarea și gestionarea obiectelor	2		
Introducere în programarea concurentă	2		
Gestionarea firelor de execuție	2		
Comunicarea între firele de execuție	2		
Mecanisme de sincronizare între fire de execuție	2		
Gestionarea fluxurilor de intrare ieșire și tratarea erorilor	2		
Implementarea aplicațiilor multifir cu interfețe grafice	2		
Gestionarea proceselor în linux	2		
Introducere în limbajul de modelare UML	2		
Diagrame structurale UML	2		
Diagrame comportamentale UML	2		
Construirea arhitecturilor proiectelor software	2		

Exemple de aplicatii de control			
Bibliografie			
1. T. Leția. Programarea avansată în Java. Editura Albastră (Microinformatica), ISBN 973-650-063-2,2002 (281 pag.).			
2. T.Leția. Programarea în limbajele Java și JavaScript. Editura Dacia, ISBN 973-35-1348-2,2002(252pag.).			
3. Clean Code Martin, R. C. (2008). Clean code: A handbook of agile software craftsmanship. Prentice Hall.			
4. Design Patterns: Elements of Reusable Object-Oriented Software Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design patterns: Elements of reusable object-oriented software. Addison-Wesley.			
5. Object-Oriented Systems Analysis and Design Using UML			
Bennett, S., McRobb, S., & Farmer, R. (2010). Object-oriented systems analysis and design using UML (4th ed.). McGraw-Hill.			
6. Fundamentals of Software Architecture Richards, M., & Ford, N. (2020). Fundamentals of software architecture: An engineering approach. O'Reilly Media.			
9.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
Prezentarea mediului de programare	1	Prezentare de exemple, discuții, aplicații practice / în caz de forță majoră, on-line platforma Teams	Prezenta obligatorie
Implementarea de aplicatii utilizand concept de baza OOP	1		
Aplicarea sabloanelor de proiectare in aplicatii OOP	1		
Aplicarea sabloanelor de proiectare OOP in aplicatii grafice	1		
Implementarea in Java a diagramelor UML	1		
Implementarea aplicatiilor concurente utilizand fire de executie	1		
Comunicarea in sistem distribuite inglobate utilizand brokerul de mesaje MQTT	2		
Implementarea de aplicatii distribuite utilizand brokerul de mesaje Apache Kafka si Rabbit MQ	2		
Familiarizarea cu librariile pentru dezvoltare de aplicatii complexe java (Spring Framework, JPA, Rest Templates) si comunicarea cu server de baze de date (Postgresql)	4		
Bibliografie			
http://control.aut.utcluj.ro:8000 Portal documentatie curs si laborator			

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

Disciplina dezvoltă abilități practice de programare concurentă și de timp real, esențiale pentru companiile care activează în domeniul ingineriei sistemelor complexe și al automatizărilor industriale. Conținutul susține direct competențele specifice programelor IAISC și ICAF, vizând proiectarea de soluții originale pentru controlul proceselor și optimizarea performanțelor software în sisteme încorporate

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare (și forma evaluare: continuă/sumativă)	11.3 Pondere din nota finală
11.4 Curs	Elaborare și prezentare proiect de materie incluzând tehnici și concepte prezentate și puse în practică la curs și laborator.	Examinare orală	50%
11.5 Seminar/Laborator /Proiect	Examinarea deprinderilor și cunoștințelor practice obținute în urma participării la laborator.	Examen practic / în caz de forță majoră, on-line platforma Teams	50%
11.6 Standard minim de performanță			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
02.07.2026	Curs	Conf. Dr. Ing. Mihai HULEA	
	Aplicații	Conf. Dr. Ing. Mihai HULEA	

Data avizării în Consiliul Departamentului de Automatică	Director Departament Automatică
_____	Prof.dr.ing. Honoriu VĂLEAN
Data aprobării în Consiliul Facultății de Automatică și Calculatoare	Decan
_____	Prof.dr.ing. Vlad MUREȘAN