

## 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	Master
1.6 Programul de studii / Calificarea	Inginerie Software / Master
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	7.20

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Limbaje și Sisteme de Tipuri</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	Prof. dr. ing. Eneia Todoran – <a href="mailto:Eneia.Todoran@cs.utcluj.ro">Eneia.Todoran@cs.utcluj.ro</a>				
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare ( E – examen, C – colocviu, V – verificare)	E
2.7 Regimul disciplinei	DA – de aprofundare, DS – de sinteza, DC – complementară				DS
	DI – Impusă, DOp – opțională, DFac – facultativă				DOp

### 3. Timpul total estimat

3.1 Număr de ore pe săptămână	3	din care:	Curs	2	Seminar	1	Laborator	-	Proiect	-
3.2 Număr de ore pe semestru	42	din care:	Curs	28	Seminar	14	Laborator	-	Proiect	-
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										10
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										15
(d) Tutoriat										10
(e) Examinări										3
(f) Alte activități:										-
3.4 Total ore studiu individual (suma (3.3(a)...3.3(f)))					58					
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	Limbaje de programare (curs nivel licența)
4.2 de competențe	Operarea cu fundamente științifice, ingineresti și matematice

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

5.1. de desfășurare a cursului	Interacțiune directă (tabla, proiector, calculator) și predare online; pentru nota maximă, participare la curs minim 70%
5.2. de desfășurare a seminarului / laboratorului / proiectului	Pentru nota maximă, participare la seminar minim 80% cu participarea activă

### 6. Competențele specifice acumulate

6.1 Competențe profesionale	<ul style="list-style-type: none"> <li>• <b>C1.3</b> - Utilizarea unor modele pentru diferite componente ale sistemelor inteligente și de viziune artificială complexe în condiții de specificare parțială</li> <li>• <b>C1.4</b> - Evaluarea formală și comparativă a caracteristicilor sistemelor inteligente și de viziune artificială complexe</li> <li>• <b>C1.5</b> - Fundamentarea teoretică a caracteristicilor sistemelor inteligente și de viziune artificială complexe proiectate, bazată pe tendințele moderne teoretice și practice</li> </ul> <p><b>C2</b> - Utilizarea tehnicii de calcul în domeniile inteligenței și viziunii artificiale și a aplicațiilor acestora.</p> <ul style="list-style-type: none"> <li>• <b>C2.1</b> - Identificarea și descrierea structurii și și a modului de funcționare al componentelor și al sistemelor inteligente și de viziune artificială</li> <li>• <b>C2.2</b> - Explicarea rolului, interacțiunilor și al caracteristicilor funcționale ale componentelor celor mai recente sisteme inteligente și de viziune artificială raportate în literatura științifică de specialitate</li> <li>• <b>C2.3</b> - Construirea unor componente originale, hardware și software, ale sistemelor inteligente și de viziune artificială, folosind algoritmi, metode de proiectare, protocoale, limbaje de programare, structuri de date, tehnologii</li> <li>• <b>C2.4</b> - Evaluarea caracteristicilor funcționale și nefuncționale ale sistemelor inteligente și de viziune artificială, pe baza unor metrici specifice</li> </ul> <p><b>C2.5</b> - Implementarea sistemelor inteligente și de viziune artificială</p>
6.2 Competențe transversale	N/A

## 7. Obiectivele disciplinei

7.1 Obiectivul general al disciplinei	Obiectivul principal al acestei discipline este de a oferi cunostinte specifice si de a pregati studentii in vederea utilizarii de modele formale si semantice in proiectarea si verificarea sistemelor de calcul. Cunostintele sunt prezentate in contextul limbajelor de programare si specificare, cu accent pe verificarea statica a tipurilor, semantica dinamica, modelarea performantei si verificarea formala a proprietatilor sistemelor de calcul.
7.2 Obiectivele specifice	Pentru atingerea acestor obiective generale studentii vor: <ul style="list-style-type: none"> <li>• Invata sa specifice si sa proiecteze formal limbaje si sisteme de calcul</li> <li>• Invata sa verifice formal proprietati ale limbajelor si sistemelor proiectate</li> <li>• Invata tehnici de proiectare si verificare a proprietatilor limbajelor si sistemelor (ex. inductie, semantica de punct fix, bisimulare, coinductie)</li> <li>• Invata sa aplice principii si paradigme avansate de proiectare</li> <li>• Studia modul in care semantica si modelele formale permit rezolvarea de probleme complexe de proiectare, modelare cantitativa, evaluare performanta</li> <li>• Urmari sa inteleaga utilitatea modelelor formale in contextul mai larg al Stiintei si Calculatoarelor prin exemple (proiectare protocoale, performanta sisteme, modele bazate pe calcul natural, etc.)</li> </ul>

## 8. Conținuturi

8.1 Curs	Nr.ore	Metode de predare	Observații
Introducere, concepte de baza	2	Curs interactiv, expunere, exemple, discutii si intrebari	Nu este cazul
Stiluri semantice, semantica operationala	2		
Untyped Lambda Calculus (ULC): sintaxa, relatie de evaluare	2		
ULC: termeni de Brouijjn (nameless terms)	2		
Simply Typed Lambda Calculus (STLC): sintaxa si relatie de tipare,	2		
STLC: proprietatile relatiei de tipare	2		
Proiectare interpretor STLC, implementare Haskell	2		

Introducere in bisimulare si semantica algebrica	2		
Bisimulare, semantica algebrica: nedeterminism si concurenta	2		
Introducere in CCS (Calculus of Communicating Systems)	2		
Bisimulare si semantica algebrica in CCS (1)	2		
Bisimulare si semantica algebrica in CCS (2)	2		
Introducere in Dependent Types (DT)	2		
Programare functionala si DT	2		
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> ) <ol style="list-style-type: none"> <li>1. B. Pierce, <i>Types and Programming Languages</i>, MIT Press, 2002.</li> <li>2. B. Pierce, <i>Advanced Topics in Types and Programming Languages</i>, MIT Press, 2005.</li> <li>3. B. Pierce et al, <i>Software Foundations</i>, <a href="https://softwarefoundations.cis.upenn.edu">https://softwarefoundations.cis.upenn.edu</a>, 2023.</li> <li>4. R. Milner. <i>Communicating and mobile systems: the pi-calculus</i>. Cambridge Univ. Press, 1999.</li> <li>5. D. Sangiorgi, <i>Introduction to Bisimulation and Coinduction</i>, Cambridge University Press, 2011.</li> <li>6. D.P. Friedman, D.T. Christiansen, <i>The Little Typer</i>, MIT Press, 2018.</li> <li>7. J.W. De Bakker, E.P. De Vink. <i>Control flow semantics</i>. MIT Press, 1996.</li> <li>8. J. Hillston, <i>A compositional approach to performance modeling</i>, Cambridge University Press, 1996.</li> <li>9. <i>Types and Programing Languages</i> (lecture notes, adapted from 1), Tehnical University of Cluj-Napoca, <a href="https://ftp.utcluj.ro/pub/users/gc/LST/tpl-2020.pdf">https://ftp.utcluj.ro/pub/users/gc/LST/tpl-2020.pdf</a>, 2020.</li> <li>10. E.N. Todoran. <i>Introducere in Semantica Limbajelor de Progmarea</i>, Note de curs si seminar, Universitatea Tehnica Cluj-Napoca, <a href="http://users.utcluj.ro/~eneia/aplc-2016.pdf">http://users.utcluj.ro/~eneia/aplc-2016.pdf</a> 2016.</li> </ol>			
<b>8.2 Aplicații (seminar/laborator/proiect)*</b>	Nr.ore	Metode de predare	
Semantica operationala structurata	2	Expunere, rezolvare probleme cu participarea activa a studentilor	Nu este cazul
Proiectare cu sisteme de tranzitie	2		
Proiectare interpretor Haskell pentru STLC	2		
Demonstrare legi CCS in semantica de bisimulare	3		
Demonstrare legi CCS in semantica de bisimulare	3		
Programarea functionala cu DT	2		
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> ) <ol style="list-style-type: none"> <li>1. B. Pierce, <i>Types and Programming Languages</i>, MIT Press, 2002.</li> <li>2. B. Pierce et al, <i>Software Foundations</i>, <a href="https://softwarefoundations.cis.upenn.edu">https://softwarefoundations.cis.upenn.edu</a>, 2023.</li> <li>3. D. Sangiorgi, <i>Introduction to Bisimulation and Coinduction</i>, Cambridge University Press, 2011.</li> <li>4. E.M. Clarke et al, <i>Handbook of Model Checking</i>, Springer, 2018.</li> <li>5. D.P. Friedman, D.T. Christiansen, <i>The Little Typer</i>, MIT Press, 2018.</li> <li>6. <i>Types and Programing Languages</i> (lecture notes, adapted from 1), Tehnical University of Cluj-Napoca, <a href="https://ftp.utcluj.ro/pub/users/gc/LST/tpl-2020.pdf">https://ftp.utcluj.ro/pub/users/gc/LST/tpl-2020.pdf</a>, 2020.</li> <li>7. E.N. Todoran. <i>Introducere in Semantica Limbajelor de Progmarea</i>, Note de curs si seminar, Universitatea Tehnica Cluj-Napoca, <a href="http://users.utcluj.ro/~eneia/aplc-2016.pdf">http://users.utcluj.ro/~eneia/aplc-2016.pdf</a> 2016.</li> </ol>			

\*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

Această disciplină introduce cunoștințe de bază în domeniile Semantica și Metode Formale. Prezentarea este realizată în contextul Limbajelor de Programare și Specificare. Limbajele și sistemele de calcul sunt descrise matematic utilizând sintaxa formală și sunt echipate cu semantica formală în funcție de context și necesitate. Din perspectiva ingineriască, această disciplină este foarte importantă pentru dezvoltarea sistemelor de calcul care impun standarde severe de calitate: fiabilitate, siguranță în funcționare, performanță măsurabilă, etc. Conținutul disciplinei este sincronizat cu ultimele avansuri în domeniu, pe baza de monografii, studii și cursuri predate la universități de prestigiu din Europa și SUA.

### 10. Evaluare

Tip activitate	Criterii de evaluare	Metode de evaluare	Pondere din nota finală
----------------	----------------------	--------------------	-------------------------

Curs	Abilitatea de rezolvare a unor probleme specifice domeniului. Prezenta, (inter)activitate in timpul orelor de curs	Evaluare finala	75%
Seminar	Abilitatea de rezolvare a unor probleme specifice domeniului. Prezenta, (inter)activitate in timpul orelor de seminar	Elaborare paper stiintific Teme seminar saptamanale	25%
Laborator	-	-	-
Proiect	-	-	-

Standard minim de performanță:

Modelarea si rezolvarea unor probleme de proiectare semantica limbaje sau sisteme, utilizand aparatul formal specific domeniului.

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
05.06.2024	Curs	Prof.dr.ing. Eneia Todoran	
	Aplicații	Prof.dr.ing. Eneia Todoran	

Data avizării în Consiliul Departamentului de 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