

## FISA DISCIPLINEI

### 1. Date despre program

1.1	Institutia de invatamint superior	Universitatea Tehnica din Cluj-Napoca
1.2	Facultatea	Automatica si Calculatoare
1.3	Departamentul	Calculatoare
1.4	Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5	Ciclul de studii	Master
1.6	Programul de studii/Calificarea	Inginerie Software/ Master
1.7	Forma de invatamint	IF – invatamant cu frecventa
1.8	Codul disciplinei	7.1

### 2. Date despre disciplina

2.1	Denumirea disciplinei	<b>Statistica si probabilitati</b>									
2.2	Aria tematica (subject area)	Calculatoare si Tehnologia Informatiei									
2.3	Responsabil de curs	Prof. Dr. Ioan Rasa, <a href="mailto:Ioan.Rasa@math.utcluj.ro">Ioan.Rasa@math.utcluj.ro</a>									
2.4	Titularul activităților de seminar / laborator / proiect	Prof. Dr. Ioan Rasa, <a href="mailto:Ioan.Rasa@math.utcluj.ro">Ioan.Rasa@math.utcluj.ro</a>									
2.5	Anul de studii	I	2.6	Semestrul	2	2.7	Evaluarea	examen	2.8	Regimul disciplinei	DS/OP

### 3. Timpul total estimat

An/ Sem	Denumirea disciplinei	Nr. sapt.	Curs			Aplicații			Stud. Ind.	TOTAL	Credit
			[ore/săpt.]			[ore/sem.]					
			S	L	P	S	L	P			
I/2	<b>Statistica si probabilitati</b>	14	2	1		28	14		58	100	4

3.1	Numar de ore pe saptamina	3	3.2	din care curs	2	3.3	aplicatii	2
3.4	Total ore din planul de inv.	42	3.5	din care curs	28	3.6	aplicatii	14
Studiul individual								Ore
Studiul dupa manual, suport de curs, bibliografie si notite								15
Documentarea suplimentara in biblioteca, pe platformele electronice si pe teren								15
Pregatire seminarii/laboratoare, teme, referate, portofolii, eseuri								15
Tutoriat								10
Examinari								3
Alte activitati								
3.7	Total ore studiul individual	58						
3.8	Total ore pe semestru	100						
3.9	Numar de credite	4						

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

4.1	De curriculum	Analiza matematica, algebra liniara, matematici speciale
4.2	De competente	Competentele disciplinelor de mai sus

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

5.1	De desfasurare a cursului	Tabla, proiector, calculator
5.2	De desfasurare a aplicatiilor	Calculatoare, software specific

### 6. Competente specifice acumulate

Competențe profesionale	<p><b>C1</b> - Operarea cu metode și modele matematice, tehnici și tehnologii specifice ingineresti și informatice avansate</p> <ul style="list-style-type: none"> <li>• <b>C1.1</b> - Cunoașterea și demonstrarea conceptelor și principiilor teoretice și practice avansate din domeniul sistemelor software</li> <li>• <b>C1.2</b> - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, unelte, etc.) pentru explicarea structurii și a modului de funcționare al celor mai recente tehnologii software, medii și sisteme de programe raportate în literatura științifică de specialitate</li> <li>• <b>C1.3</b> - Utilizarea unor modele și metode specifice pentru identificarea de componente și soluții software viabile în condiții de specificare parțială</li> <li>• <b>C1.4</b> - Evaluarea formală și comparativă a caracteristicilor metodelor, tehnicilor și modelelor de dezvoltare software, precum și a sistemelor software complexe.</li> <li>• <b>C1.5</b> - Fundamentarea teoretică a caracteristicilor sistemelor software complexe, bazată pe tendințele moderne teoretice și practice utilizate în toate etapele ciclului de dezvoltare software (specificare, analiza, proiectare, implementare, testare și integrare, validare).</li> </ul> <p><b>C2</b> - Elaborarea de tehnici, metode, și metodologii avansate în domeniul proiectării software, a mediilor și sistemelor de programe și a aplicațiilor acestora.</p> <ul style="list-style-type: none"> <li>• <b>C2.1</b> - Identificarea și descrierea structurii și a modului de funcționare a sistemelor de software complexe și a aplicațiilor dezvoltate pe baza acestora</li> <li>• <b>C2.2</b> - Exploatarea cunoștințelor de specialitate în vederea identificării și înțelegerii metodologiilor și tehnicilor de realizare a componentelor hardware și software</li> <li>• <b>C2.3</b> - Construirea unor componente software originale ale sistemelor avansate de programe, folosind algoritmi, tehnici, metode de proiectare, metodologii, protocoale, limbaje de programare, structuri de date, tehnologii și medii de programare complexe, raportate în literatura de specialitate.</li> <li>• <b>C2.4</b> - Utilizarea de metode, criterii și metrici de evaluare și selecție a metodologiilor de realizare a sistemelor software, a caracteristicilor lor funcționale și non-funcționale</li> <li>• <b>C2.5</b> - Elaborarea de proiecte software originale, implementarea, testarea și validarea acestora pe baza combinării inovative a celor raportate în literatura de specialitate</li> </ul>
Competențe transversale	N/A

### 7 Obiectivele disciplinei (reiesind din grila competențelor specific acumulate)

7.1	Obiectivul general al disciplinei	Insusirea principiilor fundamentale ale teoriei probabilitatilor și statisticii matematice
7.2	Obiectivele specifice	Capacitatea de a calcula probabilitati, de a modela în limbaj probabilistic probleme cu caracter practic și de a folosi metodele statisticii în situatii concrete

### 8. Continuturi

8.1. Curs (programa analitica)		Metode de predare	Observatii
1	Probabilitate, entropie, informatie		
2	Variabile aleatoare discrete		
3	Variabile aleatoare continue		
4	Media și dispersia		
5	Metoda celor mai mici patrate		
6	Corelatie și regresie		
7	Lanturi Markov		
8	Distributia limita. Exemple		
9	Lanturi Markov ascunse		

10	Testarea ipotezelor statistice		
11	Tehnici Bayes de estimare		
12	Familii Gaussiene		
13	Metoda verosimilitatii maxime		
14	Algoritmul EM		
8.2. Aplicatii (seminar)		Metode de predare	Observatii
1	Probabilitate, entropie, informatie		
2	Variabile aleatoare discrete, Variabile aleatoare continue		
3	Media si dispersia, Metoda celor mai mici patrate		
4	Lanturi Markov		
5	Tehnici Bayes de estimare		
6	Familii Gaussiene		
7	Algoritmul EM		
Bibliografie			
1. Ioan Rasa, Lectures on Probability Theory and Stochastic Processes, U.T.Pres 2006			
2. Ioan Rasa, Teoria Probabilitatilor si Aplicatii, ITCN 1994			
3. C.Jalobeanu, I.Rasa, Incertitudine si decizie. Statistica si probabilitati aplicate in management, U.T.Pres 2001			
4. T.K.Moon, Wynn C.Stirling, Mathematical Methods and Algorithms for Signal Processing, Prentice Hall 2000.			
5. T.T. Soong, Fundamentals of Probability and Statistics for Engineers, Wiley-Interscience, 2004.			

9. Coroborarea continuturilor disciplinei cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor, profesionale si angajatori din domeniul aferent programului

Continutul disciplinei a fost discutat cu colegii din alte departamente in scopul coroborarii cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor, profesionale si angajatori din domeniul aferent programului

## 10. Evaluare

Tip activitate	10.1	Criterii de evaluare	10.2	Metode de evaluare	10.3	Ponderea din nota finala
Curs		Capacitatea de a prezenta un subiect teoretic cu demonstratii		Lucrare scrisa (teorie)		25%
Aplicatii		Abilitatea de a rezolva probleme specifice domeniului		Lucrare scrisa (probleme)		75%
10.4 Standard minim de performanta						

Titularul de Disciplina  
Prof.dr.Ioan Rasa

Director departament  
Prof.dr.ing. Rodica Potolea

## FISA DISCIPLINEI

### 1. Date despre program

1.1	Institutia de invatamint superior	Universitatea Tehnica din Cluj-Napoca
1.2	Facultatea	Automatica si Calculatoare
1.3	Departamentul	Calculatoare
1.4	Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5	Ciclul de studii	Master
1.6	Programul de studii/Calificarea	Inginerie Software/ Master
1.7	Forma de invatamint	IF – invatamant cu frecventa
1.8	Codul disciplinei	7.2

### 2. Date despre disciplina

2.1	Denumirea disciplinei	<b>Limbaje si Sisteme de Tipuri</b>									
2.2	Aria tematica (subject area)	Calculatoare si Tehnologia Informatiei									
2.3	Responsabil de curs	Prof.dr.ing. Eneia Todoran – <a href="mailto:Eneia.Todoran@cs.utcluj.ro">Eneia.Todoran@cs.utcluj.ro</a>									
2.4	Titularul activităților de seminar / laborator / proiect	Prof.dr.ing. Eneia Todoran – <a href="mailto:Eneia.Todoran@cs.utcluj.ro">Eneia.Todoran@cs.utcluj.ro</a>									
2.5	Anul de studii	I	2.6	Semestrul	2	2.7	Evaluarea	examen	2.8	Regimul disciplinei	DS/OP

### 3. Timpul total estimat

An/ Sem	Denumirea disciplinei	Nr. sapt.	Curs			Aplicații			Stud. Ind.	TOTAL	Credit
			[ore/săpt.]			[ore/sem.]					
				S	L	P		S			
I/2	<b>Limbaje si Sisteme de Tipuri</b>	14	2	1		28	14		62	104	4

3.1	Numar de ore pe saptamina	3	3.2	din care curs	2	3.3	aplicatii	1
3.4	Total ore din planul de inv.	42	3.5	din care curs	28	3.6	aplicatii	14
Studiul individual								Ore
Studiul dupa manual, suport de curs, bibliografie si notite								25
Documentarea suplimentara in biblioteca, pe platformele electronice si pe teren								10
Pregatire seminarii/laboratoare, teme, referate, portofolii, eseuri								10
Tutoriat								10
Examinari								3
Alte activitati								
3.7	Total ore studiul individual			62				
3.8	Total ore pe semestru			104				
3.9	Numar de credite			4				

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

4.1	De curriculum	Limbaje de programare (curs nivel licenta)
4.2	De competente	Operarea cu fundamente stiintifice, ingineresti si matematice

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

5.1	De desfasurare a cursului	Tabla, proiector, calculator (prezentare interactiva); pentru nota maxima, prezenta la curs minim 70%
5.2	De desfasurare a aplicatiilor	Pentru nota maxima, prezenta la seminar minim 90% cu participarea activa

### 6. Competente specifice acumulate

Competențe profesionale	<p><b>C1</b> - Operarea cu metode și modele matematice, tehnici și tehnologii specifice ingineresti și informatice avansate</p> <ul style="list-style-type: none"> <li>• <b>C1.1</b> - Cunoașterea și demonstrarea conceptelor și principiilor teoretice și practice avansate din domeniul sistemelor software</li> <li>• <b>C1.2</b> - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, unelte, etc.) pentru explicarea structurii și a modului de funcționare al celor mai recente tehnologii software, medii și sisteme de programe raportate în literatura științifică de specialitate</li> <li>• <b>C1.3</b> - Utilizarea unor modele și metode specifice pentru identificarea de componente și soluții software viabile în condiții de specificare parțială</li> <li>• <b>C1.4</b> - Evaluarea formală și comparativă a caracteristicilor metodelor, tehnicilor și modelelor de dezvoltare software, precum și a sistemelor software complexe.</li> <li>• <b>C1.5</b> - Fundamentarea teoretică a caracteristicilor sistemelor software complexe, bazată pe tendințele moderne teoretice și practice utilizate în toate etapele ciclului de dezvoltare software (specificare, analiza, proiectare, implementare, testare și integrare, validare).</li> </ul> <p><b>C2</b> - Elaborarea de tehnici, metode, și metodologii avansate în domeniul proiectării software, a mediilor și sistemelor de programe și a aplicațiilor acestora.</p> <ul style="list-style-type: none"> <li>• <b>C2.1</b> - Identificarea și descrierea structurii și a modului de funcționare a sistemelor de software complexe și a aplicațiilor dezvoltate pe baza acestora</li> <li>• <b>C2.2</b> - Exploatarea cunoștințelor de specialitate în vederea identificării și înțelegerii metodologiilor și tehnicilor de realizare a componentelor hardware și software</li> <li>• <b>C2.3</b> - Construirea unor componente software originale ale sistemelor avansate de programe, folosind algoritmi, tehnici, metode de proiectare, metodologii, protocoale, limbaje de programare, structuri de date, tehnologii și medii de programare complexe, raportate în literatura de specialitate.</li> <li>• <b>C2.4</b> - Utilizarea de metode, criterii și metrici de evaluare și selecție a metodologiilor de realizare a sistemelor software, a caracteristicilor lor funcționale și non-funcționale</li> <li>• <b>C2.5</b> - Elaborarea de proiecte software originale, implementarea, testarea și validarea acestora pe baza combinării inovative a celor raportate în literatura de specialitate</li> </ul>
Competențe transversale	N/A

## 7 Obiectivele disciplinei (reiesind din grila competențelor specific acumulate)

7.1	Obiectivul general al disciplinei	Obiectivul principal al acestei discipline este de a oferi cunoștințe specifice și de a pregăti studenții în vederea utilizării de modele formale și semantice în proiectarea și verificarea sistemelor de calcul. Cunoștințele sunt prezentate în contextul limbajelor de programare și specificare, cu accent pe verificarea statică a tipurilor, semantica dinamică, modelarea performanței și verificarea formală a proprietăților sistemelor de calcul.
7.2	Obiectivele specifice	Pentru atingerea acestor obiective generale studenții vor: <ul style="list-style-type: none"> <li>• Invată să specifice și să proiecteze formal limbaje și sisteme de calcul</li> <li>• Invată să verifice formal proprietăți ale limbajelor și sistemelor proiectate</li> <li>• Invată tehnici de proiectare și verificare a proprietăților limbajelor și sistemelor (ex. inducție, semantica de punct fix, bisimulare, coinducție)</li> <li>• Invată să aplice principii și paradigme avansate de proiectare</li> </ul>

		<ul style="list-style-type: none"> <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. Continuturi

8.1. Curs (programa analitica)		Metode de predare	Observatii
1	Introducere, concepte de baza	Expunere la tabla, prezentare cu videoprojector, discutii	Nu este cazul
2	Semantica operationala		
3	Semantica denotationala		
4	Recursivitate si semantica de punct fix		
5	Stare, control, evaluare		
6	Tehnici semantice (continuari, monade)		
7	Domenii semantice		
8	Nedeterminism si concurenta		
9	Bisimulare, semantica algebrica		
10	Semantica de continuare pentru calcul distribuit		
11	Semantica statica, verificare tipuri		
12	Subiecte avansate: calcul distribuit si global		
13	Subiecte avansate: modelare performanta		
14	Subiecte avansate: calcul natural		
8.2. Aplicatii (seminar)		Metode de predare	Observatii
1	Semantica operationala structurata	Expuneri la tabla, explicatii suplimentare, discutii, rezolvare de probleme cu participarea activa a studentilor	Nu este cazul
2	Proiectare cu sisteme de tranzitie		
3	Semantica denotationala		
4	Domenii semantice		
5	Semantica statica, verificare tipuri		
6	Studiu de caz (calcul distribuit / calcul natural)		
Bibliografie			
1. J.W. De Bakker, E.P. De Vink. <i>Control flow semantics</i> . MIT Press, 1996.			
2. J. Hillston, <i>A compositional approach to performance modeling</i> , Cambridge University Press, 1996.			
3. J. Hillston, <i>Performance modeling</i> , <a href="http://www.inf.ed.ac.uk/teaching/courses/pm/">http://www.inf.ed.ac.uk/teaching/courses/pm/</a> , lecture notes, 2011.			
4. B. Jacobs, J. Rutten, An introduction to (co)algebras and (co)induction, In D. Sangiorgi, J. Rutten, editors, <i>Advanced topics in bisimulation and coinduction</i> , pp. 38-99, <a href="http://homepages.cwi.nl/~janr/papers/files-of-papers/2011_Jacobs_Rutten_new.pdf">http://homepages.cwi.nl/~janr/papers/files-of-papers/2011_Jacobs_Rutten_new.pdf</a> , 2011			
5. R. Milner. <i>Communicating and mobile systems: the pi-calculus</i> . Cambridge Univ. Press, 1999.			
6. B. Pierce, (Ed.). <i>Advanced topics in programming languages and type systems</i> . MIT Press, 2005.			
7. B. Pierce. <i>Programming languages and type systems</i> . MIT Press, 2002.			
8. A. Pitts. <i>Denotational semantics</i> , <a href="http://www.cl.cam.ac.uk/teaching/1112/DenotSem/dens-notes-bw.pdf">http://www.cl.cam.ac.uk/teaching/1112/DenotSem/dens-notes-bw.pdf</a> , lecture notes, 2012.			
9. F. Turbak, D. Gifford. <i>Design concepts in programming languages</i> . MIT Press, 2009.			
10. E.N. Todoran. <i>Limbaje si sisteme de tipuri</i> . Note de curs si seminar, Universitatea Tehnica Cluj-Napoca, <a href="http://users.utcluj.ro/~eneia/fl.htm">http://users.utcluj.ro/~eneia/fl.htm</a> , 2010.			
11. E.N. Todoran. <i>Limbaje si sisteme de tipuri – semantica denotationala</i> . Note de curs si seminar, Universitatea Tehnica Cluj-Napoca, <a href="http://users.utcluj.ro/~eneia/fl.htm">http://users.utcluj.ro/~eneia/fl.htm</a> , 2012.			

## 9. Coroborarea continuturilor disciplinei cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor, profesionale si angajatori din domeniul aferent programului

Aceasta disciplina introduce cunostinte de baza in domeniile Semantica si Metode Formale. Prezentarea este realizata in contextul Limbajelor de Programare si Specificare. Limbajele si sistemele de calcul sunt descrise matematic utilizand sintaxa formala si sunt echipate cu semantica

formala in functie de context si necesitate. Din perspectiva inginereasca, aceasta disciplina este foarte importanta pentru dezvoltarea sistemelor de calcul care impun standarde severe de calitate: fiabilitate, siguranta in functionare, performanta masurabila, etc. Continutul disciplinei este sincronizat cu ultimele avansuri in domeniu, pe baza de monografii, studii si cursuri predate la universitati de prestigiu din Europa si SUA. Disciplina a fost evaluata odata cu programul de studiu master in Ingineria Calculatoarelor de catre ARACIS.

## 10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Ponderea din nota finala
Curs	Abilitatea de rezolvare a unor problem specifice domeniului. Prezenta, (inter)activitate in timpul orelor de curs	Examen scris	70%
Aplicatii	Abilitatea de rezolvare a unor problem specifice domeniului. Prezenta, (inter)activitate in timpul orelor de seminar	Elaborare paper stiintific	30%
10.4 Standard minim de performanta			
Modelarea si rezolvarea unor problem de proiectare semantica limbaje sau sisteme, utilizand aparatul formal specific domeniului.			

Titularul de Disciplina  
Prof.dr.ing. Eneia Todoran

Director departament  
Prof.dr.ing. Rodica Potolea

## FISA DISCIPLINEI

### 1. Date despre program

1.1	Institutia de invatamint superior	Universitatea Tehnica din Cluj-Napoca
1.2	Facultatea	Automatica si Calculatoare
1.3	Departamentul	Calculatoare
1.4	Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5	Ciclul de studii	Master
1.6	Programul de studii/Calificarea	Inginerie Software/ Master
1.7	Forma de invatamint	IF – invatamant cu frecventa
1.8	Codul disciplinei	7.3

### 2. Date despre disciplina

2.1	Denumirea disciplinei	<b>Algoritmi si Calculabilitate</b>									
2.2	Aria tematica (subject area)	Calculatoare si Tehnologia Informatiei									
2.3	Responsabil de curs	Prof. dr. ing. Rodica Potolea, <a href="mailto:Rodica.Potolea@cs.utcluj.ro">Rodica.Potolea@cs.utcluj.ro</a>									
2.4	Titularul activităților de seminar / laborator / proiect	Prof. dr. ing. Rodica Potolea, <a href="mailto:Rodica.Potolea@cs.utcluj.ro">Rodica.Potolea@cs.utcluj.ro</a>									
2.5	Anul de studii	I	2.6	Semestrul	2	2.7	Evaluarea	examen	2.8	Regimul disciplinei	DS/OP

### 3. Timpul total estimat

An/ Sem	Denumirea disciplinei	Nr. sapt.	Curs			Aplicații			Stud. Ind.	TOTAL	Credit
			[ore/săpt.]			[ore/sem.]					
			S	L	P	S	L	P			
I/2	<b>Algoritmi si Calculabilitate</b>	14	2	1		28	14		58	100	4

3.1	Numar de ore pe saptamina	3	3.2	din care curs	2	3.3	aplicatii	1
3.4	Total ore din planul de inv.	42	3.5	din care curs	28	3.6	aplicatii	14
Studiul individual								Ore
Studiul dupa manual, suport de curs, bibliografie si notite								40
Documentarea suplimentara in biblioteca, pe platformele electronice si pe teren								
Pregatire seminarii/laboratoare, teme, referate, portofolii, eseuri								16
Tutoriat								
Examinari								2
Alte activitati								
3.7	Total ore studiul individual			58				
3.8	Total ore pe semestru			100				
3.9	Numar de credite			4				

### 4. Preconditii

4.1	De curriculum	Structuri de Date, Tehnici de Programare, Algoritmi,
4.2	De competente	Evaluare eficienta algoritmi, Cunosterea algoritmilor fundamentali pe structurile de date de baza

### 5. Conditii

5.1	De desfasurare a cursului	Tabla, proiector, calculator
5.2	De desfasurare a aplicatiilor	Calculatoare

### 6. Competente specifice acumulate



Competențe profesionale	<p><b>C1</b> - Operarea cu metode și modele matematice, tehnici și tehnologii specifice ingineresti și informatice avansate</p> <ul style="list-style-type: none"> <li>• <b>C1.1</b> - Cunoașterea și demonstrarea conceptelor și principiilor teoretice și practice avansate din domeniul sistemelor software</li> <li>• <b>C1.2</b> - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, unelte, etc.) pentru explicarea structurii și a modului de funcționare al celor mai recente tehnologii software, medii și sisteme de programe raportate în literatura științifică de specialitate</li> <li>• <b>C1.3</b> - Utilizarea unor modele și metode specifice pentru identificarea de componente și soluții software viabile în condiții de specificare parțială</li> <li>• <b>C1.4</b> - Evaluarea formală și comparativă a caracteristicilor metodelor, tehnicilor și modelelor de dezvoltare software, precum și a sistemelor software complexe.</li> <li>• <b>C1.5</b> - Fundamentarea teoretică a caracteristicilor sistemelor software complexe, bazată pe tendințele moderne teoretice și practice utilizate în toate etapele ciclului de dezvoltare software (specificare, analiza, proiectare, implementare, testare și integrare, validare).</li> </ul> <p><b>C2</b> - Elaborarea de tehnici, metode, și metodologii avansate în domeniul proiectării software, a mediilor și sistemelor de programe și a aplicațiilor acestora.</p> <ul style="list-style-type: none"> <li>• <b>C2.1</b> - Identificarea și descrierea structurii și a modului de funcționare a sistemelor de software complexe și a aplicațiilor dezvoltate pe baza acestora</li> <li>• <b>C2.2</b> - Exploatarea cunoștințelor de specialitate în vederea identificării și înțelegerii metodologiilor și tehnicilor de realizare a componentelor hardware și software</li> <li>• <b>C2.3</b> - Construirea unor componente software originale ale sistemelor avansate de programe, folosind algoritmi, tehnici, metode de proiectare, metodologii, protocoale, limbaje de programare, structuri de date, tehnologii și medii de programare complexe, raportate în literatura de specialitate.</li> <li>• <b>C2.4</b> - Utilizarea de metode, criterii și metrici de evaluare și selecție a metodologiilor de realizare a sistemelor software, a caracteristicilor lor funcționale și non-funcționale</li> <li>• <b>C2.5</b> - Elaborarea de proiecte software originale, implementarea, testarea și validarea acestora pe baza combinării inovative a celor raportate în literatura de specialitate</li> </ul>
Competențe transversale	<p>CT3 - Exersarea deprinderii de autoeducare continuă și demonstrarea de abilități critice, inovatoare și de cercetare</p>

## 7 Obiectivele disciplinei (reiesind din grila competențelor specific acumulate)

7.1	Obiectivul general al disciplinei	Analiza obiectivă a unei probleme și identificarea complexității sale; identificarea de soluții potențiale și alegerea celei potrivite contextului dat.
7.2	Obiectivele specifice	Calcul de complexitate; diferențierea claselor de complexitate; cunoșterea tipurilor de probleme care nu au soluții „usoare”.

## 8. Continuturi

8.1. Curs (programa analitică)		Metode de predare	Observatii
1	Complexitate		
2	Timp polinomial, P și NP		
3	NP completitudine		
4	Reductibilitate		
5	P=?NP		
6	Demonstrare NP-comp		

7	Probleme NP-comp		
8	Model Computational		
9	Masina Turing		
10	Reducere NP-comp1 (Circuit-SAT, SAT, 3-FNC-SAT)		
11	Reducere NP-comp2 (Clica, Acoperire varfuri, Suma)		
12	Reducere NP-comp3 (Ciclu Hamiltonian, Comis Voiajor)		
13	Aplicatii NP-comp si aproximari 1		
14	Aplicatii NP-comp si aproximari 2		
8.2. Aplicatii (seminar/lucrari/proiect)		Metode de predare	Observatii
1	P vs NPC		
2	Demonstrari apartenenta NPC		
3	Algoritmi de reducere1		
4	Algoritmi de reducere2		
5	Aproximari1		
6	Aproximari2		
7	Probleme NPC		
Bibliografie			
1. Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. <i>Introduction to Algorithms</i> . 2nd ed. Cambridge, MA: MIT Press, 2001. ISBN: 9780262032933			
2. Papadimitriou, C. H. <i>Computational Complexity</i> . 1st ed. Boston: Addison Wesley Publishing Company, 1994. ISBN: 0201530821			
3. Arora, Sanjeev and Barak, Boaz <i>Complexity Theory: A Modern Approach, Princeton University, available on the web, <a href="http://www.cs.princeton.edu/theory/index.php/Compbook/Draft">http://www.cs.princeton.edu/theory/index.php/Compbook/Draft</a></i>			
4. Sipser, Michael. <i>Introduction to the Theory of Computation</i> . 2nd ed. Boston, MA: Course Technology, 2005. ISBN: 9780534950972.			

9. Coroborarea continuturilor disciplinei cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor, profesionale si angajatori din domeniul aferent programului

Identificarea problemelor "dificile" si rezolvarea lor aproximativa; cunoasterea de solutii exacte ale problemelor "usoare"

## 10. Evaluare

Tip activitate	10.1	Criterii de evaluare	10.2	Metode de evaluare	10.3	Ponderea din nota finala
Curs		examinare		Examen scris (E)		80%
Aplicatii		Prezentare/eseu/rezolvare problem complexe		Prezentare (P)		20%
10.4 Standard minim de performanta						
Min 5 (80%E+20%P>=5)						

Titularul de Disciplina  
Prof. dr. ing. Rodica Potolea

Director departament  
Prof. dr. ing. Rodica Potolea

## FISA DISCIPLINEI

### 1. Date despre program

1.1	Institutia de invatamint superior	Universitatea Tehnica din Cluj-Napoca
1.2	Facultatea	Automatica si Calculatoare
1.3	Departamentul	Calculatoare
1.4	Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5	Ciclul de studii	Master
1.6	Programul de studii/Calificarea	Inginerie Software/ Master
1.7	Forma de invatamint	IF – invatamant cu frecventa
1.8	Codul disciplinei	8.

### 2. Date despre disciplina

2.1	Denumirea disciplinei	<b>Sisteme interactive</b>									
2.2	Aria tematica (subject area)	Calculatoare si Tehnologia Informatiei									
2.3	Responsabil de curs	Prof.dr.ing. Dorian Gorgan, <a href="mailto:dorian.gorgan@cs.utcluj.ro">dorian.gorgan@cs.utcluj.ro</a>									
2.4	Titularul activităților de seminar / laborator / proiect	Prof.dr.ing. Dorian Gorgan, <a href="mailto:dorian.gorgan@cs.utcluj.ro">dorian.gorgan@cs.utcluj.ro</a>									
2.5	Anul de studii	I	2.6	Semestrul	2	2.7	Evaluarea	examen	2.8	Regimul disciplinei	DS/OB

### 3. Timpul total estimat

An/ Sem	Denumirea disciplinei	Nr. sapt.	Curs			Aplicații			Stud. Ind.	TOTAL	Credit
			[ore/săpt.]			[ore/sem.]					
			S	L	P	S	L	P			
I/2	<b>Sisteme interactive</b>	14	2	1		28	14		58	100	4

3.1	Numar de ore pe saptamina	3	3.2	din care curs	2	3.3	aplicatii	1
3.4	Total ore din planul de inv.	42	3.5	din care curs	28	3.6	aplicatii	14
Studiul individual								Ore
Studiul dupa manual, suport de curs, bibliografie si notite								20
Documentarea suplimentara in biblioteca, pe platformele electronice si pe teren								10
Pregatire seminarii/laboratoare, teme, referate, portofolii, eseuri								21
Tutoriat								4
Examinari								3
Alte activitati								-
3.7	Total ore studiul individual			58				
3.8	Total ore pe semestru			100				
3.9	Numar de credite			4				

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

4.1	De curriculum	Programarea într-un limbaj obiectual de nivel înalt (Ex. C++, Java).
4.2	De competente	Metodologia de dezvoltare a unei aplicatii software.

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

5.1	De desfasurare a cursului	Prezență la curs minim 75% pentru admiterea la examenul final
5.2	De desfasurare a aplicatiilor	Prezență obligatorie 100% pentru admiterea la examenul final

### 6. Competente specifice acumulate

Competențe profesionale	<p><b>C1</b> - Operarea cu metode și modele matematice, tehnici și tehnologii specifice ingineresti și informatice avansate</p> <ul style="list-style-type: none"> <li>• <b>C1.1</b> - Cunoașterea și demonstrarea conceptelor și principiilor teoretice și practice avansate din domeniul sistemelor software</li> <li>• <b>C1.2</b> - Folosirea de teorii și instrumente specifice (algoritmi, scheme, modele, unelte, etc.) pentru explicarea structurii și a modului de funcționare al celor mai recente tehnologii software, medii și sisteme de programe raportate în literatura științifică de specialitate</li> <li>• <b>C1.3</b> - Utilizarea unor modele și metode specifice pentru identificarea de componente și soluții software viabile în condiții de specificare parțială</li> <li>• <b>C1.4</b> - Evaluarea formală și comparativă a caracteristicilor metodelor, tehnicilor și modelelor de dezvoltare software, precum și a sistemelor software complexe.</li> <li>• <b>C1.5</b> - Fundamentarea teoretică a caracteristicilor sistemelor software complexe, bazată pe tendințele moderne teoretice și practice utilizate în toate etapele ciclului de dezvoltare software (specificare, analiza, proiectare, implementare, testare și integrare, validare).</li> </ul> <p><b>C3</b> - Specificarea, analiza, modelarea, proiectarea, verificarea, testarea, validarea, și mentenanța sistemelor software avansate și a componentelor software, folosind instrumentele specifice domeniului</p> <ul style="list-style-type: none"> <li>• <b>C3.1</b> - Demonstrarea cunoașterii tehnologiilor, mediilor de programare, a uneltelor CASE de dezvoltare software și a conceptelor sistemelor de programe complexe</li> <li>• <b>C3.2</b> - Analiza și explicarea rolului, interacțiunilor și al modului de funcționare al componentelor software dezvoltate pe baza celor mai noi metodologii de realizare a sistemelor software complexe -propușe în literatura științifică</li> <li>• <b>C3.3</b> - Analiza, modelarea și proiectarea inovativă a sistemelor de calcul și a aplicațiilor informatice, a componentelor hardware și software aferente</li> <li>• <b>C3.4</b> - Evaluarea comparativă, sintetică, inclusiv experimentală, a alternativelor de rezolvare pentru optimizarea performanțelor, pe baza criteriilor de utilizabilitate</li> <li>• <b>C3.5</b> - Dezvoltarea și implementarea de soluții software originale pentru problemele specifice domeniului, pornind de la un set de cerințe informal specificate</li> </ul>
Competențe transversale	N/A

### 7 Obiectivele disciplinei (reiesind din grila competențelor specific acumulate)

7.1	Obiectivul general al disciplinei	Principalul obiectiv al disciplinei este dezvoltarea aplicațiilor interactive prin însușirea tehnicilor de analiză, specificare, proiectare, implementare și evaluare a componentelor care asigură interacțiunea cu utilizatorul. Se evidențiază conceptele și tehnicile din ingineria software specifice metodologiilor orientate utilizator.
7.2	Obiectivele specifice	<p>Pentru atingerea acestor obiective generale, studenții vor învăța să:</p> <ul style="list-style-type: none"> <li>• Proiecteze arhitectura sistemelor software interactive.</li> <li>• Utilizeze unelte software pentru dezvoltarea aplicațiilor interactive;</li> <li>• Desfășoare o activitate de cercetare bibliografică și experimentală, ale cărei rezultate sunt redactate într-o lucrare științifică;</li> <li>• Realizeze o sinteză și o analiză științifică și, de a susține o prezentare orală a unei teme științifice;</li> <li>• Realizeze un proiect conform metodologiei de dezvoltare și evaluare a aplicațiilor interactive orientate utilizator;</li> <li>• Lucreze individual sau în echipa.</li> </ul>

## 8. Continuturi

8.1. Curs (programa analitica)		Metode de predare	Observatii
1	Introducere. Istoric.	Expunere la tablă, prezentare cu videoproiectorul, discuții	Nu este cazul.
2	Conceptele dezvoltării interfețelor utilizator.		
3	Conceptele de comunicare intrări și ieșiri.		
4	Proiectarea orientată utilizator.		
5	Metodologia proiectării interfețelor utilizator.		
6	Utilizabilitatea în interfețele utilizator grafice.		
7	Definirea cerințelor utilizator. Descrierea și analiza taskurilor. Prototipizarea interfeței utilizator. Evaluarea interfeței utilizator.		
8	Tehnici și stiluri de interacțiune.		
9	Tehnici de interacțiune cu obiecte din spațiul virtual.		
10	Tehnici de interacțiune cu suprafețe modelate prin particule .		
11	Interfețe multimodale. Subiecte din cercetarea științifică actuală.		
12	Modele de obiecte active.		
13	Tehnologii utilizate in interfețele utilizator actuale: tehnologii Web, servicii Web, Web semantic, terminale wireless, tehnologii multimedia.		
14	Unelte, medii de lucru si limbaje utilizate pentru dezvoltarea interfețelor utilizator grafice.		
8.2. Aplicatii (seminar)		Metode de predare	Observatii
1	Conceptele dezvoltării interfețelor utilizator.	Studii de caz pe subiecte din domeniul aplicațiilor interactive, exemplificări prin utilizarea uneltelor software și a tehnologiilor specializate, expuneri la tablă, explicații suplimentare, discuții.	Nu este cazul.
2	Proiectarea orientată utilizator. Metodologia proiectării interfețelor utilizator.		
3	Definirea cerințelor utilizator. Descrierea și analiza taskurilor. Prototipizarea interfeței utilizator. Evaluarea interfeței utilizator.		
4	Tehnici și stiluri de interacțiune. Tehnici de interacțiune cu obiecte din spațiul virtual.		
5	Interfețe multimodale. Subiecte din cercetarea științifică actuală.		
6	Tehnologii utilizate in interfețele utilizator actuale: tehnologii Web, servicii Web, Web semantic, terminale wireless, tehnologii multimedia.		
7	Unelte, medii de lucru si limbaje utilizate pentru dezvoltarea interfețelor utilizator.		
<b>Bibliografie</b> <b>In biblioteca UTC-N</b> 1. B. Shneiderman, Designing the User Interface. Strategies for Effective Human Computer Interaction, Addison-Wesley, 1992. 2. A. Watt, F. Policarpo, 3D Games. Real-time Rendering and Software Technology, Addison-Wesley, 2001. <b>In bibliotecii virtuale</b> 1. Curs Sisteme Interactive, <a href="http://cgis.utcluj.ro/education/71-is">http://cgis.utcluj.ro/education/71-is</a> 2. Resurse curs Sisteme Interactive, <a href="http://cgis.utcluj.ro/didactic">http://cgis.utcluj.ro/didactic</a>			

## 9. Coroborarea continuturilor disciplinei cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor, profesionale si angajatori din domeniul aferent programului

Spre deosebire de ingineria software consacrată, cursul prezintă metodologia orientată utilizator, folosită pe scară largă în dezvoltarea aplicațiilor interactivă. Se studiază și exemplifică tehnici specifice acestei metodologii bazate pe concepul de utilizabilitate, scenarii utilizator, prototipizare, metafore, evaluare cognitivă, evaluare euristică, interacțiune multimodală etc. Conținutul disciplinei a fost discutat cu actori importanți din acest domeniu, atât din mediul academic cât și cel industrial, din România sau alte țări. Disciplina a fost evaluată de către ARACIS, odată cu alte programe de studiu de master.

## 10. Evaluare

Tip activitate	10.1	Criterii de evaluare	10.2	Metode de evaluare	10.3	Pondere din nota finala
Curs		La examenul scris (E) se verifică însușirea cunoștințelor predate la curs.		Examen scris		40%
Activitate la Curs		Activitatea la curs (AC) reflectă participarea activă la prezentările și dezbaterile științifice de la curs.		Verificări pe parcurs, discuții		10%
Aplicatii		Lucrarea științifică (L) demonstrează capacitatea de realizare a unui studiu științific și elaborarea unei lucrări sau raport științific. Proiectul (P) demonstrează abilitatea utilizării metodologiei de dezvoltare a aplicațiilor interactive.		Lucrare științifică, Proiect		50%
10.4 Standard minim de performanta						
Nota finală: $N = 0,4 \cdot E + 0,5 \cdot (L+P)/2 + 0,1 \cdot AC$						
Condiția de obținere a creditelor: $N \geq 5$ ; $E \geq 5$ ; $L \geq 5$ ; $P \geq 5$ ; $AC \geq 5$ .						

Titularul de Disciplina  
Prof. dr. ing. Dorian Gorgan

Director departament  
Prof.dr.ing. Rodica Potolea

## FISA DISCIPLINEI

### 1. Date despre program

1.1	Institutia de invatamint superior	Universitatea Tehnica din Cluj-Napoca
1.2	Facultatea	Automatica si Calculatoare
1.3	Departamentul	Calculatoare
1.4	Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5	Ciclul de studii	Master
1.6	Programul de studii/Calificarea	Inginerie Software/ Master
1.7	Forma de invatamint	IF – invatamant cu frecventa
1.8	Codul disciplinei	9.

### 2. Date despre disciplina

2.1	Denumirea disciplinei	<b>Managementul Proiectelor</b>									
2.2	Aria tematica (subject area)	Calculatoare si Tehnologia Informatiei									
2.3	Responsabil de curs	Prof.dr.ing. Mihaela Dinsoreanu- <a href="mailto:Mihaela.Dinsoreanu@cs.utcluj.ro">Mihaela.Dinsoreanu@cs.utcluj.ro</a>									
2.4	Titularul activităților de seminar / laborator / proiect	Prof.dr.ing. Mihaela Dinsoreanu- <a href="mailto:Mihaela.Dinsoreanu@cs.utcluj.ro">Mihaela.Dinsoreanu@cs.utcluj.ro</a>									
2.5	Anul de studii	I	2.6	Semestrul	2	2.7	Evaluarea	examen	2.8	Regimul disciplinei	DID/OB

### 3. Timpul total estimat

An/ Sem	Denumirea disciplinei	Nr. sapt.	Curs			Aplicații			Stud. Ind.	TOTAL	Credit
			[ore/săpt.]			[ore/sem.]					
			S	L	P	S	L	P			
I/2	<b>Managementul Proiectelor</b>	14	2	1		28	14		58	100	4

3.1	Numar de ore pe saptamina	3	3.2	din care curs	2	3.3	aplicatii	1
3.4	Total ore din planul de inv.	42	3.5	din care curs	28	3.6	aplicatii	14
Studiul individual								Ore
Studiul dupa manual, suport de curs, bibliografie si notite								20
Documentarea suplimentara in biblioteca, pe platformele electronice si pe teren								20
Pregatire seminarii/laboratoare, teme, referate, portofolii, eseuri								10
Tutoriat								5
Examinari								3
Alte activitati								0
3.7	Total ore studiul individual			58				
3.8	Total ore pe semestru			100				
3.9	Numar de credite			4				

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

4.1	De curriculum	Inginerie software
4.2	De competente	Elaborarea de tehnici, tehnologii, metode și metodologii specifice sistemelor informatice

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

5.1	De desfasurare a cursului	Prezență la curs minim 50% pentru admiterea la examenul final
5.2	De desfasurare a aplicatiilor	Prezență obligatorie 100% pentru admiterea la examenul final

### 6. Competente specifice acumulate

Competențe profesionale	<p><b>C3</b> - Specificarea, analiza, modelarea, proiectarea, verificarea, testarea, validarea, și mentenanța sistemelor software avansate și a componentelor software, folosind instrumentele specifice domeniului</p> <ul style="list-style-type: none"> <li>• <b>C3.1</b> - Demonstrarea cunoașterii tehnologiilor, mediilor de programare, a uneltelor CASE de dezvoltare software și a conceptelor sistemelor de programe complexe</li> <li>• <b>C3.2</b> - Analiza și explicarea rolului, interacțiunilor și al modului de funcționare al componentelor software dezvoltate pe baza celor mai noi metodologii de realizare a sistemelor software complexe -propușe în literatura științifică</li> <li>• <b>C3.3</b> - Analiza, modelarea și proiectarea inovativă a sistemelor de calcul și a aplicațiilor informatice, a componentelor hardware și software aferente</li> <li>• <b>C3.4</b> - Evaluarea comparativă, sintetică, inclusiv experimentală, a alternativelor de rezolvare pentru optimizarea performanțelor, pe baza criteriilor de utilizabilitate</li> <li>• <b>C3.5</b> - Dezvoltarea și implementarea de soluții software originale pentru problemele specifice domeniului, pornind de la un set de cerințe informal specificate</li> </ul> <p><b>C4</b> - Integrarea contextuală și integritatea sistemelor software complexe</p> <ul style="list-style-type: none"> <li>• <b>C4.1</b> - Demonstrarea cunoașterii și înțelegerii elementelor de interoperabilitate și integrare specifice sistemelor software, luate atât în ansamblu cât și pe module</li> <li>• <b>C4.2</b> - Folosirea unor cunoștințe interdisciplinare pentru adaptarea sistemelor software complexe în raport cu cerințele dinamice ale domeniului de aplicații</li> <li>• <b>C4.3</b> - Utilizarea combinată a unor principii și metode clasice și originale pentru integrarea componentelor unor sisteme de calcul complexe</li> <li>• <b>C4.4</b> - Folosirea standardelor de calitate, siguranță și securitate în prelucrarea informațiilor și în integrarea sistemelor software complexe.</li> <li>• <b>C4.5</b> - Realizarea de proiecte interdisciplinare, incluzând identificarea și analiza problemei, elaborarea specificațiilor, proiectarea software, implementarea testarea funcțională și evaluarea criteriilor de calitate, securitate și de performanță specifice, precum și validarea sistemului software integrat.</li> </ul>
Competențe transversale	N/A

## 7 Obiectivele disciplinei (reiesind din grila competențelor specific acumulate)

7.1	Obiectivul general al disciplinei	Principalul obiectiv al acestei discipline este de a oferi informații specifice și de a pregăti studenții în vederea gestionării corecte a proiectelor software. Astfel, se urmărește conferirea capacității de a elabora un plan de proiect corect, conform cu cerințele proiectului și de a monitoriza/controla și gestiona elaborarea proiectului.
7.2	Obiectivele specifice	<p>Pentru atingerea acestor obiective generale, studenții vor:</p> <ul style="list-style-type: none"> <li>• Urmări să înțeleagă conceptele componente ale unui plan de proiect precum și tehnicile de elaborare a planului de proiect.</li> <li>• Studia tehnici și unelte specifice diferitelor aspecte ale gestionării proiectelor (ex. gestionarea pachetelor de lucru, timpului, resurselor, costului etc)</li> <li>• Studia tehnici și unelte de estimare, monitorizare și control, analiza a riscurilor etc.</li> <li>• Învăța să analizeze specificațiile și să elaboreze un plan de proiect;</li> <li>• Învăța să monitorizeze și controleze evoluția proiectului până la finalizare aplicând metode specifice;</li> </ul>

## 8. Continuturi

8.1. Curs (programa analitica)		Metode de predare	Observatii
1	Introducere	Expunere la tablă,	
2	Disciplina Managementului de Proiect – Notiuni de baza		



3	Managementul Proiectelor in Metodologiile Agile	prezentare cu videoproiect orul, discuții		
4	Managementul Proiectelor in Procesul Unificat			
5	Rolul Managerului de Proiect			
6	Planificarea si Adaptarea Procesului			
7	Planificarea Disciplinelor			
8	Planificarea si Adaptarea Proiectului			
9	Planificarea Rolurilor de Proiect			
10	Planificarea anvergurii, WBS si planificarea in timp			
11	Planificarea Resurselor			
12	Planificarea mediului si managementul schimbarii			
13	Managementul Riscului			
14	Estimarea si Esalonarea Efortului			
8.2. Aplicatii (lucrari)			Metode de predare	Observatii
1	Planificarea si Adaptarea Procesului		Rezolvarea unor exercitii, explicații suplimentare discuții	
2	Planificarea Disciplinelor			
3	Planificarea si Adaptarea Proiectului			
4	Planificarea Rolurilor de Proiect			
5	Planificarea anvergurii, WBS si planificarea in timp			
6	Planificarea Resurselor			
7	Planificarea mediului si managementul schimbarii			
<b>Bibliografie</b>				
1. <i>The Unified Software Development Process (Hardcover)</i> Grady Booch, James Rumbaugh, Ivar Jacobson, Addison Wesley, 1998.				
2. <i>Object Solutions : Managing the Object-Oriented Project (Addison-Wesley Object Technology Series)</i> , Grady Booch, Addison Wesley, 1995				
3. <i>Software Project Management: A Unified Framework</i> , Walker Royce, Addison Wesley				
4. <i>Planning Extreme Programming</i> , Kent Beck, Addison Wesley, 2000				

## 9. Coroborarea continuturilor disciplinei cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor, profesionale si angajatori din domeniul aferent programului

Întrucât această disciplină este foarte importantă pentru gestionarea proiectelor software, conținutul ei este cât se poate de modern deoarece enunță principiile gestionării proiectelor software și prezintă apoi tehnici de gestionare a tuturor aspectelor implicate. Conținutul disciplinei a fost discutat cu actori importanți din acest domeniu, atât academici cât și industriali, din România, Europa și S.U.A. Disciplina a fost evaluată, o dată cu programul de studiu de master Inginerie Software, de către ARACIS.

## 10. Evaluare

Tip activitate	10.1	Criterii de evaluare	10.2	Metode de evaluare	10.3	Pondere din nota finala
Curs		Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de curs		Examen scris		60%
Aplicatii		Abilitatea de rezolvare a unor probleme specifice domeniului Prezență, (inter)activitate în timpul orelor de aplicatii		Rezolvarea unor probleme specifice		40%
<b>10.4 Standard minim de performanta</b>						
Planificarea si monitorizarea proiectelor software, utilizând aparatul formal specific domeniului.						

Titularul de Disciplina  
Prof.dr.ing Mihaela Dinsoreanu

Director departament  
Prof.dr.ing. Rodica Potolea

## FISA DISCIPLINEI

### 1. Date despre program

1.1	Institutia de invatamint superior	Universitatea Tehnica din Cluj-Napoca
1.2	Facultatea	Automatica si Calculatoare
1.3	Departamentul	Calculatoare
1.4	Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5	Ciclul de studii	Master
1.6	Programul de studii/Calificarea	Inginerie Software/ Master
1.7	Forma de invatamint	IF – invatamant cu frecventa
1.8	Codul disciplinei	10.

### 2. Date despre disciplina

2.1	Denumirea disciplinei	<b>Proiectare software sensibil la context</b>									
2.2	Aria tematica (subject area)	Calculatoare si Tehnologia Informatiei									
2.3	Responsabil de curs	Conf. Dr. ing. Anca Rarău- <a href="mailto:Anca.Rarau@cs.utcluj.ro">Anca.Rarau@cs.utcluj.ro</a>									
2.4	Titularul activităților de seminar / laborator / proiect	Conf. Dr. ing. Anca Rarău- <a href="mailto:Anca.Rarau@cs.utcluj.ro">Anca.Rarau@cs.utcluj.ro</a>									
2.5	Anul de studii	I	2.6	Semestrul	2	2.7	Evaluarea	examen	2.8	Regimul disciplinei	DS/OB

### 3. Timpul total estimat

An/ Sem	Denumirea disciplinei	Nr. sapt.	Curs			Aplicații			Stud. Ind.	TOTAL	Credit		
			[ore/săpt.]			[ore/sem.]							
				S	L	P		S				L	P
I/2	<b>Proiectare software sensibil la context</b>	14	2		1		28		14		58	100	4

3.1	Numar de ore pe saptamina	3	3.2	din care curs	2	3.3	aplicatii	1
3.4	Total ore din planul de inv.	42	3.5	din care curs	28	3.6	aplicatii	14
Studiul individual								Ore
Studiul dupa manual, suport de curs, bibliografie si notite								20
Documentarea suplimentara in biblioteca, pe platformele electronice si pe teren								10
Pregatire seminarii/laboratoare, teme, referate, portofolii, eseuri								10
Tutoriat								16
Examinari								2
Alte activitati								0
3.7	Total ore studiul individual			58				
3.8	Total ore pe semestru			100				
3.9	Numar de credite			4				

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

4.1	De curriculum	n/a
4.2	De competente	Operarea cu fundamente științifice, ingineresti și ale informaticii

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

5.1	De desfasurare a cursului	Prezență la curs minim 50% pentru admiterea la examenul final.
5.2	De desfasurare a aplicatiilor	Prezență obligatorie 100% pentru admiterea la examenul final.

### 6. Competente specifice acumulate

Competențe profesionale	<p><b>C4 - Integrarea contextuală și integritatea sistemelor software complexe</b></p> <ul style="list-style-type: none"> <li>• <b>C4.1</b> - Demonstrarea cunoașterii și înțelegerii elementelor de interoperabilitate și integrare specifice sistemelor software, luate atât în ansamblu cât și pe module</li> <li>• <b>C4.2</b> - Folosirea unor cunoștințe interdisciplinare pentru adaptarea sistemelor software complexe în raport cu cerințele dinamice ale domeniului de aplicații</li> <li>• <b>C4.3</b> - Utilizarea combinată a unor principii și metode clasice și originale pentru integrarea componentelor unor sisteme de calcul complexe</li> <li>• <b>C4.4</b> - Folosirea standardelor de calitate, siguranță și securitate în prelucrarea informațiilor și în integrarea sistemelor software complexe.</li> <li>• <b>C4.5</b> - Realizarea de proiecte interdisciplinare, incluzând identificarea și analiza problemei, elaborarea specificațiilor, proiectarea software, implementarea testarea funcțională și evaluarea criteriilor de calitate, securitate și de performanță specifice, precum și validarea sistemului software integrat.</li> </ul> <p><b>C5 - Îmbinarea creativă a cunoștințelor multidisciplinare din domeniul științei calculatoarelor și tehnologiei informației în vederea cercetării, specificării, proiectării, optimizării, implementării, testării și evaluării de teorii, algoritmi, tehnici, metode și metodologii originale specifice sistemelor software complexe.</b></p> <ul style="list-style-type: none"> <li>• <b>C5.1</b> - Demonstrarea cunoașterii metodologiei de cercetare, proiectare, implementare, optimizare și testare a sistemelor de calcul complexe</li> <li>• <b>C5.2</b> - Demonstrarea capacității de a analiza și interpreta situații noi prin prisma cunoștințelor fundamentale din domeniul calculatoarelor și tehnologiei informației</li> <li>• <b>C5.3</b> - Îmbinarea creativă, bazată pe descoperirea de legături semantice și funcționale noi, a diferite principii de proiectare moderne din domeniul calculatoarelor și tehnologiei informației pentru rezolvarea unor probleme de optimizare</li> <li>• <b>C5.4</b> - Fundamentarea activității de cercetare și proiectare inovativă din domeniul calculatoarelor pe criterii corecte de evaluare</li> <li>• <b>C5.5</b> - Realizarea de activități de cercetare cu finalitate practică demonstrată prin prototipuri software și / sau hardware funcționale.</li> </ul>
Competențe transversale	N/A

### 7 Obiectivele disciplinei (reiesind din grila competențelor specific acumulate)

7.1	Obiectivul general al disciplinei	Principalul obiectiv al acestei discipline este de a introduce conceptele de baza ale unui domeniu care este în expansiune – sistemele sensibile la context. Astfel se urmărește asigurarea bagajului de cunoștințe ce poate fi folosit la evaluarea critică a soluțiilor de proiectare existente, precum și la propunerea de noi soluții.
7.2	Obiectivele specifice	Pentru atingerea obiectivului general, studenții vor studia soluțiile existente de proiectare ale sistemelor sensibile la context pornind de la principiile arhitecturale și ajungându-se până la paradigmele de programare și limbajele specifice. De asemenea, studenții vor analiza și proiecta un sistem de dimensiuni mari în care vor aplica principiile și paradigmele învățate.

### 8. Continuturi

8.1. Curs (programa analitică)		Metode de predare	Observații
1	Introducere. Motivația apariției sistemelor sensibile la context. Definirea cerințelor proiectului referitor la orașul inteligent sensibil la context.	Expunere la tablă, prezentare cu video-proiectorul, discuții.	Nu este cazul.
2	Reprezentarea contextului: probleme de modelare – calitate, cantitate, completitudine și raționare.		
3	Reprezentarea contextului: clase de contexte – perechi atribut valoare, scheme de marcare, bazate pe logică, orientate obiect și bazate pe ontologii.		
4	Arhitectura sistemelor sensibile la context: Generalități.		
5	Arhitectura sistemelor sensibile la context: SOCAM, ContextToolkit.		

6	Arhitectura sistemelor sensibile la context: ubi-UCAM, MobiPADS.		
7	Principiile sistemelor sensibile la context: ortogonalității, separării, conservării ordinii temporale și comportării complet definite.		
8	Protocoale de descoperire a contextului: R-CDP, Directed Diffusion, SCI.		
9	Limbaje de programare sensibile la context: Generalități.		
10	Limbaje de programare sensibile la context: Aware C#.		
11	Limbaje de programare sensibile la context: COP.		
12	Limbaje de programare sensibile la context: ContextL.		
13	Sisteme autonome / Calcul autonom.		
14	Prezentarea, pe echipe, a proiectelor referitoare la orașul inteligent senzitiv la context.		
8.2. Aplicații (lucrări)		Metode de predare	Observatii
1	Cum se citește un articol de cercetare?		
2	Discutarea temelor de prezentare primite de studenți (articolele fac parte din următoarele categorii: starea actuala a domeniului, arhitectura sisteme sensibile la context, testare sisteme sensibile la context, paradigme de programare sensibilă la context, studii de caz).	Expuneri la tablă, prezentare cu video-proiectorul, explicații suplimentare, discuții.	Nu este cazul.
3	Prezentări și discuții referitoare la starea actuala a domeniului.		
4	Prezentări și discuții referitoare la arhitectura sistemelor sensibile la context.		
5	Prezentări și discuții referitoare la testarea sistemelor sensibile la context.		
6	Prezentări și discuții referitoare la paradigme de programare sensibilă la context.		
7	Discutarea proiectării unui oraș inteligent sensibil la context.		
<b>Bibliografie</b> 1. A. Rarău, M. Cremene. K-I. Bența, Sistem senzitive la context, Editura Albastră, 2008. 2. S. Loke, L. Loke, Context Aware Pervasive Systems: The Architecture of a New Breed of Applications, CRC Press, 2006.			

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

În cadrul studiilor de licență, studenții aprofundează proiectarea a multor clase de sisteme printre care se numără sistemele distribuite, sistemele paralele etc. Această disciplină își propune să completeze varietatea de sisteme acoperite în timpul studiilor de licență, propunând studierea unei clase de sisteme software complexe care devine din ce în ce mai prezentă atât în aria de cercetare cât și în domeniul comercial.

## 10. Evaluare

Tip activitate	10.1	Criterii de evaluare	10.2	Metode de evaluare	10.3	Ponderea din nota finală
Curs		Abilitatea de a propune soluții unor probleme specifice domeniului. Prezență, (inter)activitate în timpul orelor de curs.		Examen scris.		50%
Aplicații		Abilitatea de a identifica probleme și de a evalua critic soluțiile existente în domeniu și de a propune îmbunătățiri. Prezență, (inter)activitate în timpul orelor.		Prezentări articole și proiect.		50%
<b>10.4 Standard minim de performanță</b>						
Înțelegerea conceptelor de bază ale domeniului și demonstrarea abilității de a evalua critic soluțiile existente.						

Titularul de Disciplina  
Conf. Dr. Ing. Anca Rarau

Director departament  
Prof.dr.ing. Rodica Potolea

## FISA DISCIPLINEI

### 1. Date despre program

1.1	Institutia de invatamint superior	Universitatea Tehnica din Cluj-Napoca
1.2	Facultatea	Automatica si Calculatoare
1.3	Departamentul	Calculatoare
1.4	Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5	Ciclul de studii	Master
1.6	Programul de studii/Calificarea	Inginerie Software / Master
1.7	Forma de invatamint	IF – invatamant cu frecventa
1.8	Codul disciplinei	11.

### 2. Date despre disciplina

2.1	Denumirea disciplinei	<b>Proiect IS 2</b>									
2.2	Aria tematica (subject area)	Calculatoare si Tehnologia Informatiei									
2.3	Responsabil de curs	Prof. dr. ing. Ioan Salomie - <a href="mailto:Ioan.Salomie@cs.utcluj.ro">Ioan.Salomie@cs.utcluj.ro</a>									
2.4	Titularul activităților de seminar / laborator / proiect	Prof.dr.ing.Gorgan Dorian - <a href="mailto:Dorian.Gorgan@cs.utcluj.ro">Dorian.Gorgan@cs.utcluj.ro</a> Prof.dr.ing. Potolea Rodica- <a href="mailto:Potolea.Rodica@cs.utcluj.ro">Potolea.Rodica@cs.utcluj.ro</a> Prof.dr.ing. Eneia Todoran – <a href="mailto:Eneia.Todoran@cs.utcluj.ro">Eneia.Todoran@cs.utcluj.ro</a> Prof.dr.ing. Mihaela Dinsoreanu - <a href="mailto:Mihaela.Dinsoreanu@cs.utcluj.ro">Mihaela.Dinsoreanu@cs.utcluj.ro</a>									
2.5	Anul de studii	I	2.6	Semestrul	2	2.7	Evaluarea	colocviu	2.8	Regimul disciplinei	DS/OB

### 3. Timpul total estimat

An/ Sem	Denumirea disciplinei	Nr. sapt.	Curs			Aplicații			Stud. Ind.	TOTAL	Credit
			[ore/săpt.]			[ore/sem.]					
			S	L	P	S	L	P			
I/2	<b>Proiect IC 2</b>	14			2			28	72	100	4

3.1	Numar de ore pe saptamina	2	3.2	din care curs	-	3.3	aplicatii	2
3.4	Total ore din planul de inv.	28	3.5	din care curs	-	3.6	aplicatii	28
Studiul individual								Ore
Studiul dupa manual, suport de curs, bibliografie si notite								20
Documentarea suplimentara in biblioteca, pe platformele electronice si pe teren								20
Pregatire seminarii/laboratore, teme, referate, portofolii, eseuri								20
Tutoriat								10
Examinari								2
Alte activitati								0
3.7	Total ore studiul individual	72						
3.8	Total ore pe semestru	100						
3.9	Numar de credite	4						

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

4.1	De curriculum	Proiect IS 1
4.2	De competente	Aferente disciplinei de mai sus

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

5.1	De desfasurare a cursului	Nu este cazul
5.2	De desfasurare a aplicatiilor	Echipe si programe specifice temei de proiect

### 6. Competente specifice acumulate

Competențe profesionale	<p><b>C4</b> - Integrarea contextuală și integritatea sistemelor software complexe</p> <ul style="list-style-type: none"> <li>• <b>C4.1</b> - Demonstrarea cunoașterii și înțelegerii elementelor de interoperabilitate și integrare specifice sistemelor software, luate atât în ansamblu cât și pe module</li> <li>• <b>C4.2</b> - Folosirea unor cunoștințe interdisciplinare pentru adaptarea sistemelor software complexe în raport cu cerințele dinamice ale domeniului de aplicații</li> <li>• <b>C4.3</b> - Utilizarea combinată a unor principii și metode clasice și originale pentru integrarea componentelor unor sisteme de calcul complexe</li> <li>• <b>C4.4</b> - Folosirea standardelor de calitate, siguranță și securitate în prelucrarea informațiilor și în integrarea sistemelor software complexe</li> <li>• <b>C4.5</b> - Realizarea de proiecte interdisciplinare, incluzând identificarea și analiza problemei, elaborarea specificațiilor, proiectarea software, implementarea testarea funcțională și evaluarea criteriilor de calitate, securitate și de performanță specifice, precum și validarea sistemului software integrat</li> </ul>
Competențe transversale	N/A

### 7 Obiectivele disciplinei (reiesind din grila competențelor specific acumulate)

7.1	Obiectivul general al disciplinei	Dezvoltarea de competențe și abilități pentru elaborarea de proiecte din domeniul ingineriei software, calculatoarelor și tehnologiei informației
7.2	Obiectivele specifice	<p>Asimilarea de cunoștințe și abilități privind:</p> <ul style="list-style-type: none"> <li>• elaborarea arhitecturii unei aplicații din domeniul ingineriei software</li> <li>• proiectarea componentelor de baza ale aplicației</li> <li>• elaborarea documentației de proiectare</li> </ul>

### 8. Continuturi

8.1. Curs (programa analitica)		Metode de predare	Observatii
1	Nu e cazul		
8.2. Aplicatii (proiect)		Metode de predare	Observatii
1	Introducere	Prezentare metodologii de proiectare, Verificări periodice	
2	Elaborarea arhitecturii: - Identificarea principalelor componente		
3	Elaborarea arhitecturii: - Identificarea și stabilirea interfețelor		
4	Elaborarea arhitecturii: - Identificarea și stabilirea aspectelor dinamice		
5	Prezentarea arhitecturii elaborate (document)		
6	Proiectare: - Detalierea componentelor arhitecturale (1)		
7	Proiectare: - Detalierea componentelor arhitecturale (2)		
8	Proiectare: - Detalierea componentelor arhitecturale (3)		
9	Proiectare: - Detalierea componentelor arhitecturale (4)		
10	Proiectare: - Detalierea componentelor arhitecturale (5)		
11	Proiectare: - Detalierea componentelor arhitecturale (6)		
12	Proiectare: - Detalierea componentelor arhitecturale (7)		
13	Prezentarea proiectului (document)		
14	Elaborarea și prezentarea documentației finale		

### Bibliografie

- [1] R.S. Pressman - Software Engineering, A Practitioner's Approach, McGraw-Hill, 7/e, 2009
- [2] Ian Sommerville – Software Engineering, Addison Wesley, 8/e (2006), 9/e (2011)
- [3] Eric Braude, Michael Bernstein, Software Engineering – Modern Approaches, Wiley 2010
- [4] Capers Jones - Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies, McGraw-Hill, 2010
- [5] Peter Eeles, Peter Cripps - The Process of Software Architecting, Addison Wesley, 2010

9. Coroborarea continuturilor disciplinei cu asteptarile reprezentantilor comunitatii epistemice, asociatiilor, profesionale si angajatori din domeniul aferent programului

--

### 10. Evaluare

Tip activitate	10.1	Criterii de evaluare	10.2	Metode de evaluare	10.3	Ponderea din nota finala
Curs		Nu este cazul				
Aplicatii		Pe baza rezultatelor practice si a referatului elaborat		Evaluare orala Evaluare referat		60% 40%
10.4 Standard minim de performanta						
nota 5						

Titularul de Disciplina  
Prof.dr.ing.Ioan Salomie

Director departament  
Prof.dr.ing. Rodica Potolea

## FISA DISCIPLINEI

### 1. Date despre program

1.1	Institutia de invatamint superior	Universitatea Tehnica din Cluj-Napoca
1.2	Facultatea	Automatica si Calculatoare
1.3	Departamentul	Calculatoare
1.4	Domeniul de studii	Calculatoare si Tehnologia Informatiei
1.5	Ciclul de studii	Master
1.6	Programul de studii/Calificarea	Inginerie Software / Master
1.7	Forma de invatamint	IF – invatamant cu frecventa
1.8	Codul disciplinei	12.

### 2. Date despre disciplina

2.1	Denumirea disciplinei	<b>Activitate de cercetare 2</b>									
2.2	Aria tematica (subject area)	Calculatoare si Tehnologia Informatiei									
2.3	Responsabil de curs	Nu e cazul									
2.4	Titularul activităților de seminar / laborator / proiect	Nu e cazul									
2.5	Anul de studii	I	2.6	Semestrul	2	2.7	Evaluarea	A/R	2.8	Regimul disciplinei	DS/OB

### 3. Timpul total estimat

An/ Sem	Denumirea disciplinei	Nr. sapt.	Curs			Aplicații			Stud. Ind.	TOTAL	Credit
			[ore/săpt.]			[ore/sem.]					
				S	L	P		S			
I/2	<b>Activitate de cercetare 2</b>	14						250	250	10	

3.1	Numar de ore pe saptamina		3.2	din care curs	-	3.3	aplicatii	
3.4	Total ore din planul de inv.		3.5	din care curs	-	3.6	aplicatii	
Studiul individual								Ore
Studiul dupa manual, suport de curs, bibliografie si notite								100
Documentarea suplimentara in biblioteca, pe platformele electronice si pe teren								100
Pregatire seminarii/laboratoare, teme, referate, portofolii, eseuri								30
Tutoriat								15
Examinari								5
Alte activitati								0
3.7	Total ore studiul individual	250						
3.8	Total ore pe semestru	250						
3.9	Numar de credite	10						

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

4.1	De curriculum	Activitatea de cercetare 1
4.2	De competente	Competentele disciplinei de mai sus

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

5.1	De desfasurare a cursului	Nu este cazul
5.2	De desfasurare a aplicatiilor	Echipeamente si programe specifice temei de proiect

### 6. Competente specifice acumulate



Competențe profesionale	<p><b>C3</b> - Specificarea, analiza, modelarea, proiectarea, verificarea, testarea, validarea, și mentenanța sistemelor software avansate și a componentelor software, folosind instrumentele specifice domeniului</p> <ul style="list-style-type: none"> <li>• <b>C3.1</b> - Demonstrarea cunoașterii tehnologiilor, mediilor de programare, a uneltelor CASE de dezvoltare software și a conceptelor sistemelor de programe complexe</li> <li>• <b>C3.2</b> - Analiza și explicarea rolului, interacțiunilor și al modului de funcționare al componentelor software dezvoltate pe baza celor mai noi metodologii de realizare a sistemelor software complexe -propușe în literatura științifică</li> <li>• <b>C3.3</b> - Analiza, modelarea și proiectarea inovativă a sistemelor de calcul și a aplicațiilor informatice, a componentelor hardware și software aferente</li> <li>• <b>C3.4</b> - Evaluarea comparativă, sintetică, inclusiv experimentală, a alternativelor de rezolvare pentru optimizarea performanțelor, pe baza criteriilor de utilizabilitate</li> <li>• <b>C3.5</b> - Dezvoltarea și implementarea de soluții software originale pentru problemele specifice domeniului, pornind de la un set de cerințe informal specificate</li> </ul> <p><b>C4</b> - Integrarea contextuală și integritatea sistemelor software complexe</p> <ul style="list-style-type: none"> <li>• <b>C4.1</b> - Demonstrarea cunoașterii și înțelegerii elementelor de interoperabilitate și integrare specifice sistemelor software, luate atât în ansamblu cât și pe module</li> <li>• <b>C4.2</b> - Folosirea unor cunoștințe interdisciplinare pentru adaptarea sistemelor software complexe în raport cu cerințele dinamice ale domeniului de aplicații</li> <li>• <b>C4.3</b> - Utilizarea combinată a unor principii și metode clasice și originale pentru integrarea componentelor unor sisteme de calcul complexe</li> <li>• <b>C4.4</b> - Folosirea standardelor de calitate, siguranță și securitate în prelucrarea informațiilor și în integrarea sistemelor software complexe</li> <li>• <b>C4.5</b> - Realizarea de proiecte interdisciplinare, incluzând identificarea și analiza problemei, elaborarea specificațiilor, proiectarea software, implementarea testarea funcțională și evaluarea criteriilor de calitate, securitate și de performanță specifice, precum și validarea sistemului software integrat</li> </ul>
Competențe transversale	N/A

### 7 Obiectivele disciplinei (reiesind din grila competențelor specific acumulate)

7.1	Obiectivul general al disciplinei	Dezvoltarea de abilitați și competențe de cercetare și proiectare în domeniul ingineriei software, calculatoarelor și tehnologiei informațiilor
7.2	Obiectivele specifice	<p>Asimilarea de cunoștințe și abilitați privind:</p> <ul style="list-style-type: none"> <li>• elaborarea schemei generale sau a arhitecturii sistemului software ce urmează a fi dezvoltat</li> <li>• efectuarea de experimente, teste și verificări</li> <li>• enunțarea unor ipoteze de lucru și validarea acestora prin experimente</li> <li>• proiectarea componentelor unui sistem aplicativ</li> </ul>

### 8. Continuturi

8.1. Curs (programa analitică)		Metode de predare	Observatii
1	Nu e cazul.		
8.2. Aplicații (seminar/lucrări/proiect)		Metode de predare	Observatii
1	Definirea obiectivelor activității de cercetare pe care o va realiza în	Lucru	

	lucrarea de dizertatie; Stabileste programul de cercetare teoretica, experimentală și/sau prin simulare numerică pe care îl va realiza în lucrarea de dizertatie; Documentare asupra temei de dizertatie; Realizarea unui raport de sinteză a activitatilor derulate.	individual și verificări periodice	
Bibliografie Se stabilește de către fiecare îndrumător de proiect de dizertatie în parte.			

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

--

#### 10. Evaluare

Tip activitate	10.1	Criterii de evaluare	10.2	Metode de evaluare	10.3	Pondere din nota finală
Curs		Nu este cazul				
Aplicații		Pe baza rezultatelor practice și a referatului elaborat		Evaluare orală Evaluare referat		60% 40%
10.4 Standard minim de performanță						
nota 5						

Titularul de Disciplina  
Îndrumătorii de disertație

Director departament  
Prof. dr. ing. Rodica Potolea