

### FIȘA DISCIPLINEI

<b>Course title/ Titlul cursului:</b>	Ingineria programării				
<b>Course code/ Codul cursului:</b>	<b>Type of course/ Tipul cursului:</b>	<b>Level of course/ Nivelul cursului:</b>	<b>Year of study/ An de studiu:</b>	<b>Semester/ Semestru:</b>	<b>Number of credits/ Număr de credite:</b>
38060243	Obligatoriu	Studii de licență	3	5	5
<b>Name of lecturer/ Numele profesorului:</b>	<b>Titular curs/seminar/laborator/proiect:</b> Lector univ. Dr. Mircea I. Neamțu				
<b>Department/ Departament (ce coordonează disciplina):</b>	DEPARTAMENTUL DE MATEMATICĂ SI INFORMATICĂ				
<b>Extinderea disciplinei în planul de învățământ</b>					
<b>Lecture/Curs</b>	<b>Seminar</b>	<b>Lab/Laborator</b>	<b>Project/Proiect</b>	<b>Total (NOAD<sub>sem</sub>)</b>	
2 x 14 = 28 ore		2 x 14 = 28 ore		56	
<b>Bugetul de timp pentru studiu individual</b>					
<b>Type of activities / Denumirea activității</b>	<b>Hours/Nr. Ore</b>	<b>Type of activities / Denumirea activității</b>		<b>Hours/Nr. Ore</b>	
1. Studierea notițelor de curs	5	5. Pregătirea seminariilor/laboratoarelor		10	
2. Studiul după suport de curs	10	6. Elaborarea referatelor, temelor individuale		10	
3. Studierea bibliografiei minimale	20	7. Pregătirea pentru evaluările periodice		20	
4. Documentare suplimentară (internet, bibliotecă ș.a.)	5	8. Participarea la consultații		4	
<b>Total ore alocate studiului individual NOSI<sub>sem</sub></b>				84	
<b>Objectives of the course/ Obiectivele cursului:</b>	Formarea conceptului de <i>Ingineria programării orientată spre agent</i> (IPOA). Folosirea arhitecturilor pentru mediile de dezvoltare fiind abordate elemente de paralelism, interactivitate, viteză de reacție, adaptabilitate, interoperabilitate, comportament inteligent.				
<b>Prerequisites/ Discipline precursoare obligatorii:</b>	Tehnologii Web				
<b>Course contents/ Conținutul cursului:</b>	Cursul 1	<i>Problematika și contextul.</i> Analiza conținutului cursului (obiective; aria tematică; granularitate; trăsături specifice). TI în societatea postindustrială. Factori de influență (paradigmele întreprinderii: distribuie și autoorganizare; reinginerie). „Piața informațională”. Salturi paradigmatic: A) de la <i>produs</i> (material, local, stabil, de masă) la <i>serviciu</i> (intelectual, global, dinamic, individualizat); B) de la <i>planificare</i> (specificație în mediu închis) la <i>intenție</i> (negociere în mediu deschis).			
	Cursul	<i>Reflectarea în TI.</i> A) Paradigma <i>client-prestator</i> . de la produs-			

	2	program (bazat pe <i>obiect</i> ) spre serviciu diversificat (bazat pe <i>proces</i> ). B) Paradigma <i>calculul ca interacțiune</i> : de la program (determinist, bazat pe <i>algoritm</i> ) spre scenariu (nedeterminist, bazat pe <i>interacțiune</i> ). Implicațiile WWW (stratul de mijloc din rețele, genuri de paralelism, e-aplicațiile). Trăsături cheie: incertitudine, timp, mediu dinamic deschis, omul în buclă.
	Cursul 3	<i>Reactivitate prin excepții</i> . a) <i>Finalitate</i> : de la „salvarea programului” la „răspuns flexibil la stimulii din mediu”. b) <i>Arhitectură</i> : reluare flexibilă, adaptare la context, propagare dinamică, tratare structurată. c) <i>Structură</i> : declarare, cadru generator, semnalare, rutină de tratare).
	Cursul 4	<i>Interfețe antropocentrice</i> . Concepte (multimodal, perceptual, agent de interfață); principii (utilizabilitate, ergonomie industrială și cognitivă). Mașina ca interactant (potențial și preferințe). Elemente de dezvoltare.
	Cursul 5	<i>Ingineria programării convenționale</i> . Evoluție și abordări. Principii. Macrocaracteristici (robustețe, ergonomie, protecție, adaptabilitate, eficiență). Ciclul de viață (de la conceptualizare la extindere). Spațiu de proiectare (dimensiuni și aspecte). <i>Premise, criterii, etape</i> . Adaptare (e-aplicație, sistem moștenit, reinginerie etc.), resurse (non)preemptive, restricții, strategii.
	Cursul 6	<i>Ingineria programării orientată spre agent</i> (IPOA). Cerințe arhitecturale pentru mediile de dezvoltare: paralelism, interactivitate, viteză de reacție, adaptabilitate, interoperabilitate, comportament inteligent. Limitele și inadecvarea paradigmatelor convenționale (mai ales critica <i>orientării spre obiect</i> și a <i>algoritmului</i> ).
	Cursul 7	<i>Medii pentru programare distribuită</i> . Paralelismul ca instrument. Memorie comună și timp partajat (inclusiv reflectarea lor în sistemul de operare prin excludere mutuală). Suficiența mediilor orientate spre obiect („comunicare tehnologică” sincronizată prin secțiuni critice, mutex, semafoare). Limitările (acceptabile) ale dezvoltării entităților de <i>programare paralelă distribuită</i> prin fire implementate ca obiecte.
	Cursul 8	<i>Medii pentru programare concurentă</i> . Paralelismul ca model al lumii. Reactivitate și proactivitate, nedeterminism, timp real (inclusiv reflectarea lor în sistemul de operare prin triada întreruperi + preemptiune + mod nucleu). Interacțiune = comunicare (scop) + sincronizare (mijloc). Evenimente și așteptări, priorități dinamice, relații de filiație. Insuficiența mediilor orientate spre obiect (inadecvarea tratării proceselor prin reducționismul obiectual, ineficiența cronică). Limitările (inacceptabile) ale dezvoltării entităților de <i>programare paralelă concurentă</i> prin fire implementate ca obiecte. Corolar: nevoia de acces direct la primitive adecvate.
	Cursul 9	<i>Limbaje și medii</i> . Probleme și categorii (tradiție/calitate; program/script; interpretare/compilare/JIT; imperativ /descriptiv; expresivitate/rigoare/ortogonalitate). Trăsături: grad de OO (metode/funcții), tipizare (statică/dinamică, tare/slabă), utilizabilitate multi-platăformă, genericitate, integrabilitate (mai ales cu limbaje de nivel mai scăzut), reflectare. Abordare contextuală (exemple: „garbage collection”; „pointer”; supraîncărcare metode/operatori). Relația limbaj/mediu (ilustrare prin comparația <i>Java/Python</i> , respectiv <i>JADE/Spyse</i> ).
	Cursul 10	<i>Eficacitate, eficiență, acceptanță</i> . Concepte, nuanțe, deosebiri; subiectiv și obiectiv. Tipuri de eficiență: economică; în raport cu o

		resursă; în raport cu o echipă. Metrica performanței. Separabilitate. Evaluări calitative și cantitative. <i>Paradigma schimbării</i> . Implicații (ilustrare prin trecerea de la "client/server" la "P2P"). Relativizarea specificațiilor. Adaptare prin metoda prototipurilor (interfețelor) succesive. Variante.
	Cursul 11	<i>Beneficiari și utilizatori finali</i> . Elemente de ingineria utilizabilității (adaptate la sisteme antropocentrice). Modele cognitive. Profilarea utilizatorului (inclusiv, limitele paletelor). Modele de comportament (de la Sheridan și Rasmussen la internaut și netățean). Rolul agentului de interfață. Implicarea utilizatorilor în dezvoltare; metodele: sociotehnică; scandinavă; etnografică.
	Cursul 12	<i>Verificare și validare</i> . Testare (în raport cu <i>specificațiile</i> ). Infrastructura testării. Reutilizabilitate. Cazuri limită („patologice”). Evaluare (automată, empirică, (ne)formalizată). <i>Validare</i> (în raport cu <i>așteptările</i> ). Tipuri de validare: in ovo, in vitro, in vivo. De la validarea interfeței la validarea ecologică. Metode: variante de test Turing. Compararea metodelor de implicare a utilizatorilor (sindromul „ostaticului”).
	Cursul 13	<i>Dezvoltarea aplicațiilor de prestare de servicii, bazate pe interacțiune</i> . Implicațiile noilor paradigme. Contopiri de etape ale ciclului de viață. Echipa de dezvoltare. Metoda proiectantului șef; conducere prin excepții. Instrumente software specifice. Cele cinci grade de ignoranță (Armour). Sindromul „programului 90% gata”. De la produs-program la cunoaștere activă. Metode și metodologii software.
	Cursul 14	<i>Tendința principală: interacțiune intensă în medii deschise, dinamice, inteligente</i> . Concepte: atitudine intențională, ipoteze (simbolic și subsimbolic), euristică; agentitate (trăsături slabe și tari). Scenarii nedeterministe, organizații virtuale (exemplu: "Grid"), autoorganizare și emergență (rețele neurale artificiale, algoritmi genetici, control stigmergic). Concluzii privind alegerea mediilor de programare.
<b>Lab/Seminary contents/ Conținutul laboratorului/seminarului:</b>	Laborator 1	<i>Igiena TI</i> . Noxele informatice: cauzele (mediul, configurația, complexitatea). Ergonomie (industrială și cognitivă). Protecția utilizatorului și ergonomie în operare (accent pe monitor).
	Laborator 2	<i>Omogenizare și personalizare</i> . Verificarea deprinderilor de programare. Dezbateră preferințelor.
	Laborator 3	<i>Familiarizarea cu Windows XP</i> (accent pe Office). Critica deprinderilor (ex.: tastatura „românească”, abuzul de butoane în fragmente de program irelevante).
	Laborator 4	<i>Întreruperi și excepții</i> . Exemple de întreruperi uzuale și reflectarea lor în programe. Excepții simple (recuperare din erori, interacțiune cu utilizatorul). Critica tratării în Java. Comparație cu C#.
	Laborator 5	<i>Excepții avansate</i> . Interfața cu mediul, simularea tratării structurate, propagarea spre entitatea apelantă. Mimarea excepțiilor definite de utilizator.
	Laborator 6	<i>Interacțiunea cu aplicația</i> . Interfețe multimediale și multimodale simple (bazate numai pe facilitățile configurațiilor uzuale).
	Laborator 7	<i>Personalizarea interfețelor</i> . Jocuri de echipă cu scenarii ad-hoc și cu alternarea rolurilor de proiectant și de utilizator. Validarea (calitativă) a interfeței.
	Laborator 8	<i>Programare distribuită</i> (multitasking). Exemple de paralelism pentru viteză. Fire de execuție slab comunicante prin memorie comună, sincronizare implicită (apelare orientată spre obiect a serviciilor). Exemple în: PHP, Java/C#.

	Laborator 9	<i>Programare concurentă</i> (multithreading). Exemple de paralelism pentru timp real. Fire de execuție. Apelarea funcțiilor API și verificarea acceptării/efectuării serviciului. Comunicare prin memorie comună. Sincronizare prin evenimente cu setare manuală și așteptări limitate. Exemple cu apelarea directă a funcțiilor API (Win32).
	Laborator 10	<i>Mecanisme și tehnici generale de programare</i> . Structuri (adevare date/cod, cu exemplificare în Delphi). Recursivitate. Entități de programare autonome (fire sau procese). Comunicare (transfer, partajare, relația volum-viteză) și sincronizare (sensuri, parteneri, momente, căi). Filiație (moștenire: drepturi și obiecte).
	Laborator 11	<i>Sisteme de operare pentru medii avansate</i> . Trăsături de bază: preempțiune; mod de lucru protejat; întreruperi. Strategii de alocare a unității centrale. Prestare de servicii prin interfața de proiectare a aplicațiilor (API). Biblioteci legate dinamic.
	Laborator 12	<i>Windows XP ca dublu exemplu</i> . A) <i>Sistem de operare preemptiv</i> : gestiunea obiectuală a resurselor; priorități; gestiunea memoriei; interfața. B) <i>Aplicație în timp real</i> (ilustrarea interacțiunii fire-procese-ferestre-om-mediu).
	Laborator 13	<i>Dezvoltarea unei aplicații simple</i> . Analiza resurselor (numai pe configurații uzuale). Stabilirea cerințelor și schițarea specificațiilor. Repetarea ciclului programare-testare-validare (calitativă, a interfeței; accent pe cazurile patologice). Simularea validării ecologice.
	Laborator 14	<i>Colocviu</i> (examen preliminar bazat pe evaluarea susținerii unei lucrări de laborator, la alegere).
<b>Teaching methods/ Metode de predare:</b>		<b>Language of instruction/ Limba de predare:</b>
Prelegerea intensificată, Expunere intuitivă, Termeni/cheie		<b>Româna</b>
<b>Assesment methods/ Sisteme de evaluare:</b>	Activități aplicative - 25%	1. Teme de curs/pondere= 5 % (nCPC) 2. Referate de disciplină= 10% (nCPC) 3. Lucrări practice= 10% (CPC)
	Proiect - 25%	CPE (CPE – condiționează participarea la examen)
	Examen parțial - 25%	(nCPE – nu condiționează participarea la examen)
	Examen de semestru - 25%	(condiționează evaluarea finală)
<b>Competențe specifice disciplinei</b>		
<b>1. Competențe privind cunoașterea și înțelegerea</b>	Potențialul de a valorifica la nivel de înțelegere (nu de memorare) competențele dobândite prin/la disciplinele din amonte din planul de învățământ.	
<b>2. Competențe în domeniul explicării și interpretării</b>	Prioritizarea cognitivă netă pe următoarele categorii: 1) gândire (în contextul societății moderne); 2) înțelegere (accent pe reducerea complexității cognitive și nu a celei structurale); 3) învățare (cu accent pe deprinderi nu memorare, de exemplu, evaluarea la calculator și nu prin examen scris.	
<b>3. Competențe instrumental - aplicative</b>	Deprinderea lucrului cu sistemul de operare Windows XP, interfețe multimediale, limbaje procedurale din familia Java, C și de tip script (PHP, Python).	
<b>4. Competențe atitudinale</b>	Perspective noi privind TI în contextul globalizării din societatea post industrială bazată pe paradigma „calculul ca interacțiune” (de exemplu trecerea de la paradigma client-server la P2P).	

Competențele generale sunt menționate în Fișa specializării	
<b>Recommended reading/ Referințe bibliografice recomandate (max. 10):</b>	<ol style="list-style-type: none"><li>1. <i>Ingineria programarii</i> - Curs pe suport electronic. Mircea Iosif Neamtu</li><li>2. <i>Ingineria Programarii</i>, Florin Leon, Bucuresti, 2001</li><li>3. <i>Vulnerabilitati ale sistemelor informatice</i>, Neamtu I. Mircea, Sibiu, 2013</li></ol>
<b>More references/ Referințe bibliografice suplimentare:</b>	<ol style="list-style-type: none"><li>1. <a href="http://www.w3c.org">www.w3c.org</a></li><li>2. TENDINTE ACTUALE IN PROIECTAREA SI DEZVOLTAREA APLICATIILOR WEB, Sabin Buraga, Editura: Matrixrom [2006]</li><li>3. TEHNOLOGII XML, Sabin Buraga, Editura: Polirom [2006]</li><li>4. <i>Open source &amp; Free Software</i>, Neamtu I. Mircea, Sibiu, 2007</li></ol>
<b>Regulamentul disciplinei</b>	

Data elaborării:  
25.09.2013

Titulari disciplină:  
Lector univ. dr. Mircea Iosif Neamtu