Universitatea „Lucian Blaga” din Sibiu
Facultatea de Ştiinţe
Catedra de Informatică
Domeniul de studii de licenţă: Informatică
Specializarea: Informatică

PROGRAMA ANALITICĂ

Denumirea disciplinei: Ingineria programării
Codul disciplinei: 3906F05O043
Anul de studiu şi semestrul în care se studiază disciplina: III/5
Regimul disciplinei (obligatorie O, opţională A sau facultativă L): O
Categoria formativă (fundamentală Fd, de specialitate Sp, generală Gen): Sp
Discipline anterioare cerute *:
Forma de evaluare (examen E, verificare V, colocviu C): E
Catedra care coordonează disciplina: Catedra de Informatică
Titularul / titularii disciplinei: Conf. dr. Ioan Pop

* disciplinele studiate anterior a căror cunoaştere este necesară pentru însuşirea disciplinei

Extinderea disciplinei în planul de învăţământ *:				
Curs	Seminar	Laborator	Proiect	Total ($NOAD_{sem}$)
28		28		56

* numărul semestrial de ore de activităţi didactice directe

Bugetul de timp şi creditele alocate disciplinei			
$NOAD_{sem}$	$NOSI_{sem}$	$NOT_{sem} = NOAD_{sem} + NOSI_{sem}$	**Numărul de credite**
56	84	140	5

Obiectivele disciplinei
Obiectivele cursului a) Dezvoltarea de aplicaţii simple conform tendinţelor din TI (tehnologia informaţiei), pe baza paradigmei „Calculul ca interacţiune”. b) Adaptarea la cerinţele societăţii postindustriale prin mutarea ponderii de la produse-program (eficienţă, funcţionalitate, testare, ciclu de viaţă) spre aplicaţii prestatoare de servicii (acceptanţă, interfaţă, validare, disponibilitate). c) Extinderea acestei abordări la ingineria dezvoltării aplicaţiilor prin mutarea accentului de la cantitate, specificaţii exacte (formalizate) şi imuabile, experienţa proiectantului, performanţe (de moment), probe de recepţie etc. spre calitate, cerinţe fuzzy (neformalizate) şi modificabile, aşteptările utilizatorului, metode (de durată), asistenţă continuă. d) Simularea elaborării şi gestiunii unui proiect în condiţiile lumii reale). e) Tendinţe în ingineria programării (IP): saltul paradigmatic de la *obiect* spre *agent*.

Obiectivele activităților aplicative
(seminar, laborator, proiect)

a) Dezvoltarea de aplicații simple conform tendințelor din TI (tehnologia informației), pe baza paradigmei „Calculul ca interacțiune".
b) Adaptarea la cerințele societății postindustriale prin mutarea ponderii de la produse-program (eficiență, funcționalitate, testare, ciclu de viață) spre aplicații prestatoare de servicii (acceptanță, interfață, validare, disponibilitate).
c) Extinderea acestei abordări la ingineria dezvoltării aplicațiilor prin mutarea accentului de la cantitate, specificații exacte (formalizate) şi imuabile, experiența proiectantului, performanțe (de moment), probe de recepție etc. spre calitate, cerințe fuzzy (neformalizate) şi modificabile, aşteptările utilizatorului, metode (de durată), asistență continuă.
d) Simularea elaborării şi gestiunii unui proiect în condițiile lumii reale).
e) Tendințe în ingineria programării (IP): saltul paradigmatic de la *obiect* spre *agent*.

Conținutul disciplinei (capitolele cursului / tematica seminarului / lucrărilor practice / etapele proiectului)

CURS

Nr. crt.	Tema	Nr.ore	Săptămâna
1.	*Problematica şi contextul*. Analiza conținutului cursului (obiective; aria tematică; granularitate; trăsături specifice). TI în societatea postindustrială. Factori de influență (paradigmele întreprinderii: distribuire şi autoorganizare; reinginerie). „Piața informațională". Salturi paradigmatice: A) de la *produs* (material, local, stabil, de masă) la *serviciu* (intelectual, global, dinamic, individualizat); B) de la *planificare* (specificație în mediu închis) la *intenție* (negociere în mediu deschis).	2	1
2.	*Reflectarea în TI*. A) Paradigma *client-prestator*: de la produs-program (bazat pe *obiect*) spre serviciu diversificat (bazat pe *proces*). B) Paradigma *calculul ca interacțiune*: de la program (determinist, bazat pe *algoritm*) spre scenariu (nedeterminist, bazat pe *interacțiune*). Implicațiile WWW (stratul de mijloc din rețele, genuri de paralelism, e-aplicațiile). Trăsături cheie: incertitudine, timp, mediu dinamic deschis, omul în buclă.	2	2
3.	*Reactivitate prin excepții*. a) *Finalitate*: de la „salvarea programului" la „răspuns flexibil la stimulii din mediu". b) *Arhitectură*: reluare flexibilă, adaptare la context, propagare dinamică, tratare structurată. c) *Structură*: declarare, cadru generator, semnalare, rutină de tratare).	2	3
4.	*Interfețe antropocentrice*. Concepte (multimodal, perceptual, agent de interfață); principii (utilizabilitate, ergonomie industrială şi cognitivă). Maşina ca interactant (potențial şi preferințe). Elemente de dezvoltare.	2	4
5.	*Ingineria programării convențională*. Evoluție şi abordări. Principii. Macrocaracteristici (robustețe, ergonomie, protecție, adaptabilitate, eficiență). Ciclul de viață (de la conceptualizare	2	5

	la extindere). Spaţiul de proiectare (dimensiuni şi aspecte). *Premise, criterii,etape*. Adaptare (e-aplicaţie, sistem moştenit, reinginerie etc.), resurse (non)preemptive, restricţii, strategii.		
6.	*Ingineria programării orientată spre agent* (IPOA). Cerinţe arhitecturale pentru mediile de dezvoltare: paralelism, interactivitate, viteză de reacţie, adaptabilitate, interoperabilitate, comportament inteligent. Limitele şi inadecvarea paradigmelor convenţionale (mai ales critica *orientării spre obiect* şi a *algoritmului*).	2	6
7.	*Medii pentru programare distribuită*. Paralelismul ca instrument. Memorie comună şi timp partajat (inclusiv reflectarea lor în sistemul de operare prin excluziune mutuală). Suficienţa mediilor orientate spre obiect („comunicare tehnologică" sincronizată prin secţiuni critice, mutex, semafoare). Limitările (acceptabile) ale dezvoltării entităţilor de *programare paralelă distribuită* prin fire implementate ca obiecte.	2	7
8.	*Medii pentru programare concurentă*. Paralelismul ca model al lumii. Reactivitate şi proactivitate, nedeterminism, timp real (inclusiv reflectarea lor în sistemul de operare prin triada întreruperi + preempţiune + mod nucleu). Interacţiune = comunicare (scop) + sincronizare (mijloc). Evenimente şi aşteptări, priorităţi dinamice, relaţii de filiaţie. Insuficienţa mediilor orientate spre obiect (inadecvarea tratării proceselor prin reducţionismul obiectual, ineficienţa cronică). Limitările (inacceptabile) ale dezvoltării entităţilor de *programare paralelă concurentă* prin fire implementate ca obiecte. Corolar: nevoia de acces direct la primitive adecvate.	2	8
9.	*Limbaje şi medii*. Probleme şi categorii (tradiţie/calitate; program/script; interpretare/compilare/JIT; imperativ /descriptiv; expresivitate/rigoare/ortogonalitate). Trăsături: grad de OO (metode/funcţii), tipizare (statică/dinamică, tare/slabă), utilizabilitate multi-platformă, genericitate, integrabilitate (mai ales cu limbaje de nivel mai scăzut), reflectare. Abordare contextuală (exemple: "garbage collection"; "pointer"; supraîncărcare metode/operatori). Relaţia limbaj/mediu (ilustrare prin comparaţia *Java/Python*, respectiv *JADE/Spyse*).	2	9
10.	*Eficacitate, eficienţă, acceptanţă*. Concepte, nuanţe, deosebiri; subiectiv şi obiectiv. Tipuri de eficienţă: economică; în raport cu o resursă; în raport cu o echipă. Metrica performanţei. Separabilitate. Evaluări calitative şi cantitative. *Paradigma schimbării.* Implicaţii (ilustrare prin trecerea de la "client/server" la "P2P"). Relativizarea specificaţiilor. Adaptare prin metoda prototipurilor (interfeţelor) succesive. Variante.	2	10
11.	*Beneficiari şi utilizatori finali*. Elemente de ingineria utilizabilităţii (adaptate la sisteme antropocentrice). Modele cognitive. Profilarea utilizatorului (inclusiv, limitele paletei).	2	11

	Modele de comportament (de la Sheridan şi Rasmunsen la internaut şi netățean). Rolul agentului de interfață. Implicarea utilizatorilor în dezvoltare; metodele: sociotehnică; scandinavă; etnografică.		

12.	*Verificare şi validare*. Testare (în raport cu *specificațiile*). Infrastructura testării. Reutilizabilitate. Cazuri limită („patologice"). Evaluare (automată, empirică, (ne)formalizată). *Validare* (în raport cu *aşteptările*). 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").	2	12
13.	*Dezvoltarea aplicațiilor de prestare de servicii, bazate pe interacțiune*. Implicațiile noilor paradigme. Contopiri de etape ale ciclului de viață. Echipa de dezvoltare. Metoda proiectantului şef; conducere prin excepții. Instrumente software specifice. Cele cinci grade de ignoranță (Armour). Sindromul „programului 90% gata". De la produs-program la cunoaştere activă. Metode şi metodologii software.	2	13
14.	*Tendința principală: interacțiune intensă în medii deschise,dinamice, inteligente*. Concepte: atitudine intențională, ipoteze (simbolic şi subsimbolic), euristică; agentitate (trăsături slabe şi tari). Scenarii nedeterministe, organizații virtuale (exemplu: "Grid"), autoorganizare şi emergență (rețele neurale artificiale, algoritmi genetici, control stigmergic). Concluzii privind alegerea mediilor de programare.	2	14

LABORATOR

Nr. crt.	Tema	Nr.ore	Săptămâna
1.	*Igiena TI*. Noxele informatice: cauzele (mediul, configurația, complexitatea). Ergonomie (industrială şi cognitivă). Protecția utilizatorului şi ergonomie în operare (accent pe monitor).	2	1
2.	*Omogenizare şi personalizare.* Verificarea deprinderilor de programare. Dezbaterea preferințelor.	2	2
3.	*Familiarizarea cu Windows XP* (accent pe Office). Critica deprinderilor (ex.: tastatura „românească", abuzul de butoane In fragmente de program irelevante).	2	3
4.	*Întreruperi şi excepții*. Exemple de întreruperi uzuale şi reflectarea lor în programe. Excepții simple (recuperare din erori, interacțiune cu utilizatorul). Critica tratării în Java. Comparație cu C#.	2	4
5.	*Excepții avansate*. Interfața cu mediul, simularea tratării structurate, propagarea spre entitatea apelantă. Mimarea excepțiilor definite de utilizator.	2	5
6.	*Interacțiunea cu aplicația.* Interfețe multimediale şi multimodale simple (bazate numai pe facilitățile configurațiilor uzuale).	2	6

7.	*Personalizarea interfețelor*. Jocuri de echipă cu scenarii ad-hoc şi cu alternarea rolurilor de proiectant şi de utilizator. Validarea (calitativă) a interfeței.	2	7

8.	*Programare distribuită* (multitasking). Exemple de paralelism pentru viteză. Fire de execuție slab comunicante prin memorie comună, sincronizare implicită (apelare orientată spre obiect a serviciilor). Exemple în Java/C#.	2	8
9.	*Programare concurentă* (multithreading). Exemple de paralelism pentru timp real. Fire de execuție. Apelarea funcțiilor API şi verificarea acceptării/efectuării serviciului. Comunicare prin memorie comună. Sincronizare prin evenimente cu setare manuală şi aşteptări limitate. Exemple cu apelarea directă a funcțiilor API (Win32).	2	9
10.	*Mecanisme şi tehnici generale de programare*. Structuri (adecvare date/cod, cu exemplificare în Delphi). Recursivitate. Entități de programare autonome (fire sau procese?). Comunicare (transfer, partajare, relația volum-viteză) şi sincronizare (sensuri, parteneri, momente, căi). Filiație (moştenire: drepturi şi obiecte).	2	10
11.	*Sisteme de operare pentru medii avansate*. 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.	2	11
12.	*Windows XP ca dublu exemplu.* A) *Sistem de operare preemptiv*: gestiunea obiectuală a resurselor; priorități; gestiunea memoriei; interfața. B) *Aplicație în timp real* (ilustrarea interacțiunii fire-procese-ferestre-om-mediu).	2	12
13.	*Dezvoltarea unei aplicații simple*. Analiza resurselor (numai pe configurații uzuale). Stabilirea cerințelor şi schițarea specificațiilor. Repetarea ciclului programare-testare-validare (calitativă, a interfeței; accent pe cazurile patologice). Simularea validării ecologice.	2	13
14.	*Colocviu* (examen preliminar bazat pe evaluarea susținerii unei lucrări de laborator, la alegere).	2	14

Descrierea metodelor de predare
- Pentru predare se va folosi prelegerea, dezbaterea, învățarea prin cooperare, explicația, tematizarea. - Pentru seminar se folosesc lucrări de laborator disponibile studenților, la cerere pe suport magnetic. - mediile de programare *JADE* şi *Spyse*, CBuilder. - Sunt valabile regulamentele oficiale ale universității în legătură cu prezenta - studenților la activitățile didactice şi cu cazurile de copiat şi plagiat. - Promovarea examenului este condiționată de predarea completă a lucrărilor de proiect. - Prezenta la orele de proiect este obligatorie.

Descrierea formelor şi metodelor de evaluare a cunoştinţelor
Nota finală se va stabili după cum urmează: xxviii) Activitatea de laborator şi proiect 90 % xxix) Examen final 10 %

Bibliografie obligatorie
6. Bărbat, B.-E, S.C. Negulescu. *Bazele sistemelor în timp real*. (Capitolele 1-5 în format electronic.) 7. Bărbat, B.-E. *Sisteme inteligente orientate spre agent*. Ed. Academiei Române, Bucureşti, 2002. (Capitolele/subcapitolele: 2, 6.2, 7.1-7.3, 8.1, 8.2, 8.4, 8.5.2, 8.5.3, 9.1.1, 9.1.3, 9.5).

Bibliografie opţională
1. Richter, J.M. *Programming Applications for Microsoft Windows.* Fourth Edition. Microsoft Programming Series. Microsoft Press, Redmond WA, 1999. 2. Microsoft Corporation. *Windows 2000 (XP). Online support and information*. Hipertext (în cadrul ajutorului interactiv al sistemului), 2000-2006. 3. TIOBE Software BV. *TIOBE Company*. http://www.tiobe.com/company/index 4. TIOBE Software BV. *TIOBE Programming Community Index for November 2006*. http://www.tiobe.com/tpci. htm 5. Wikipedia, the free encyclopedia. *Comparison of programming languages*. http://en.wikipedia.org/wiki/Comparison_of_programming_languages (cu modificările de la 22:49, 6 noiembrie 2006). 6. AgentLink III. *Agent based computing. AgentLink Roadmap: Overview and Consultation Report*. University of Southampton, sept. 2005. http://www.agentlink.org/roadmap/al3rm.pdf 7. FIPA TC Agent Management. *FIPA Agent Management Specification*. Standard SC00023K (2004/18/03). http://www.fipa.org/specs/fipa00023/SC00023K.pdf

Data elaborării:

Titularul / titularii disciplinei,
Conf. Dr. Ioan Pop