

FISA DISCIPLINEI

DENUMIREA DISCIPLINEI				TEHNICI AVANSATE DE INGINERIA PROGRAMĂRII				COD: MOC1FG01	
CICLUL DE STUDII (L-licență/M-master/D-doctorat) ȘI ANUL DE STUDIU (1,2,3,4)				M 1	Semestrul	1	STATUTUL DISCIPLINEI (OB-obligatorie/OP-opțională/F-facultativă)		OP
NUMĂRUL ORELOR PE SAPTĂMÂNĂ				TOTAL ORE SEMESTRU	TOTAL ORE ACTIVITATE INDIVIDUALA	NUMĂR DE CREDITE	TIPUL DE EVALUARE (P-pe parcurs, C-colocviu, E-examen, M-mixt)	LIMBA DE PREDARE	
C	S	L	Pr.						
2	-	2	-	56	124	8	M	Română	

TITULARUL ACTIVITĂȚILOR DE CURS	GRADUL DIDACTIC ȘI ȘTIINȚIFIC, PRENUMELE, NUMELE	DEPARTAMENTUL
	LECTOR DR. ADRIAN IFTENE	INFORMATICĂ

TITULARUL ACTIVITĂȚILOR DE SEMINAR/L.P.	GRADUL DIDACTIC ȘI ȘTIINȚIFIC, PRENUMELE, NUMELE	DEPARTAMENTUL
	LECTOR DR. ADRIAN IFTENE	INFORMATICĂ

DISCIPLINE ABSOLVITE ANTERIOR	Obligatorii: Ingineria Programării Recomandate: Tehnici de Proiectare și Analiză a Algoritmilor, Tehnologii Web, Tehnici Avansate de Programare, Dezvoltarea de Aplicații pe Platforma .Net
-------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

OBIECTIVE	Dezvoltarea unei înțelegeri mai profunde a domeniului ingineriei programării. Studenții se vor familiariza cu concepte de proiectarea sistemelor complexe, arhitecturi moderne și vor face cunoștință cu abordările actuale folosite în modelarea software-ului competitiv.
-----------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

COMPETENȚE SPECIFICE ACUMULATE

COMPETENȚE PROFESIONALE	Programarea în limbaje de nivel înalt. Dezvoltarea și întreținerea aplicațiilor informatice. Proiectarea și gestiunea bazelor de date.
-------------------------	----------------------------------------------------------------------------------------------------------------------------------------

COMPETENȚE TRANSVERSALE	Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatică de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse
-------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CONTINUTUL CURSULUI	1. Ingineria programării – gestionarea proceselor de dezvoltare a software-ului, gestionarea configurațiilor aplicațiilor. Unelte și metode folosite. 2. Programarea bazată pe aspecte (AOP), pe monitorizare (MOP). 3. Dezvoltarea și întreținerea sistemelor: dezvoltarea folosind arhitecturi bazate pe servicii (SOA), dezvoltarea condusă de model (MDD, MDA), dezvoltare agilă condusă de model (AMDD), dezvoltare condusă de teste (TDD). Refactorizare. 4. Modelare, modelarea afacerilor: BPMN, JBPM, XPD.L. 5. Calitatea Serviciilor (QoS), Verificarea la execuție.
---------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

BIBLIOGRAFIE (SELECTIVĂ)	Hunt, A., Thomas, D. (1999) The Pragmatic Programmer: From Journeyman to Master, ISBN 0-201-61622-X Schmidt, D.C. (2006). "Model-Driven Engineering". IEEE Computer 39 (2). Beydeda, S., Book, M., Gruhn, V. (2005) Model-Driven Software Development. Springer. Kleppe, A., Warmer, J., and Bast Addison Wesley,., W. (2003) MDA Explained: The Model Driven Architecture (TM): Practice and Promise. 192p. OMG (2003) MDA Guide Version 1.0.1. Editors: Joaquin Miller and Jishnu Mukerji. Balzer, Y. (2004) Improve your SOA project plans, IBM, 16 July. Bell, M. (2008) "Introduction to Service-Oriented Modeling". Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley & Sons. pp. 3. ISBN 978-0-470-14111-3. SWEBOK: Guide to the Software Engineering Body of Knowledge, http://www.swebok.org/
--------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

CONȚINUTUL LUCRĂRILOR DE SEMINAR/LABORATOR	Studenții vor lucra la un proiect de cercetare ținând cont de aplicarea principiilor de bună practică din ingineria programării. La partea de implementare vor folosi Refactorizare (cu scopul de a îmbunătăți arhitectura codului existent), Testare automată. Pentru modelare vor folosi Limbaje specifice domeniu (DSL) sau cadre de lucru precum Eclipse Modeling Framework, Open Architecture Ware (OAW), etc.
--------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

BIBLIOGRAFIE (SELECTIVĂ)	Hunt, A., Thomas, D. (1999) The Pragmatic Programmer: From Journeyman to Master, ISBN 0-201-61622-X Schmidt, D.C. (2006). "Model-Driven Engineering". IEEE Computer 39 (2). Beydeda, S., Book, M., Gruhn, V. (2005) Model-Driven Software Development. Springer. Kleppe, A., Warmer, J., and Bast Addison Wesley,., W. (2003) MDA Explained: The Model Driven Architecture (TM): Practice and Promise. 192p. OMG (2003) MDA Guide Version 1.0.1. Editors: Joaquin Miller and Jishnu Mukerji. Balzer, Y. (2004) Improve your SOA project plans, IBM, 16 July. Bell, M. (2008) "Introduction to Service-Oriented Modeling". Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley & Sons. pp. 3. ISBN 978-0-470-14111-3. SWEBOK: Guide to the Software Engineering Body of Knowledge, http://www.swebok.org/
--------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

REPERE METODOLOGICE	Prezentarea teoretică a conceptelor, urmată de ilustrare practică. Experimentare la laborator.	
EVALUARE	metodele	Activitate de laborator, proiecte în echipă, test scris
	forme	Laborator: teme săptămânale de laborator în echipă Proiect: teme la nivel de subgrupă de cercetare Examen scris: se va da în penultima săptămână din semestru
	pondera formelor de evaluare în formula notei finale	Punctajul final se stabilește ca sumă a punctajelor obținute la laborator, la proiect și la examen. Ponderele notei finale va fi în jur de 20 % laboratorul, 50 % proiectul și 30 % examenul. Studentul care participă la examen, va primi o notă, altfel va fi considerat absent. Dacă unul din criteriile de promovare nu este îndeplinit, studentul va obține o nota mai mică sau egală cu 4.
	standardele minime de performanță	Punctaj minim laborator: 50% din punctajul maxim ce poate fi obținut la laborator (minim 50 % proiect) Punctaj minim examen: 40% din punctajul maxim ce poate fi obținut la examen