

FISA DISCIPLINEI

DENUMIREA DISCIPLINEI		METODE FORMALE ÎN INGINERIA SOFTWARE					COD:		
CICLUL DE STUDII (L-licență/M-master/D-doctorat) ȘI ANUL DE STUDIU (1,2,3,4)			M 2	Semestrul	I	STATUTUL DISCIPLINEI (OB-obligatorie/OP-opțională/F-facultativă)		OB	
NUMĂRUL ORELOR PE SĂPTĂMÂNĂ		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, Engleză

TITULARUL ACTIVITĂȚILOR DE CURS	GRADUL DIDACTIC ȘI ȘTIINȚIFIC, PRENUMELE, NUMELE	DEPARTAMENTUL
	PROF. DR. DOREL LUCANU	Informatică

TITULARUL ACTIVITĂȚILOR DE SEMINAR/L.P.	GRADUL DIDACTIC ȘI ȘTIINȚIFIC, PRENUMELE, NUMELE	DEPARTAMENTUL
	PROF. DR. DOREL LUCANU POSTDOC. DR. TRAIAN ȘERBĂNUȚĂ	Informatică

DISCIPLINE ABSOLVITE ANTERIOR	Logică, Programare orientată-obiect, Ingineria programării.
-------------------------------	---

OBIECTIVE*	Însușirea și înțelegerea utilizării metodelor formale în ingineria software.
------------	--

COMPETENȚE SPECIFICE ACUMULATE	
--------------------------------	--

COMPETENȚE PROFESIONALE**	<p>Cunoștințe: Semantica limbajelor de programare, logica Floyd-Hoare, contracte în POO, <i>separation logic</i>, logică temporală, verificarea programelor, model-checking, analiză de cod bazată pe demonstrare automată sau asistată.</p> <p>Abilități: Abilitatea de a formaliza într-un limbaj logic ceea ce specificarea unui program, abilitatea de a utiliza sisteme deductive și algoritmi de model-checking pentru verificarea și analiza de programe, abilitatea de a utiliza un <i>framework</i> dedicat definițiilor de limbaje de programare, abilitatea de a scrie programe mai fiabile, abilitatea de găsi defecte în programe.</p>
---------------------------	---

COMPETENȚE TRANSVERSALE	Disciplina are un puternic caracter interdisciplinar. Presupune cunoașterea cel puțin a unui limbaj de programare imperativă și a unui limbaj de programare orientată-obiect. Utilizează instrumente învățate la logică și algoritmică. Are o influență puternică în modul în care sunt gândite și scrise programele.
-------------------------	---

CONTINUTUL CURSULUI	Introducere în metodele formale. Introducere în semantica limbajelor de programare. Logica Floyd-Hoare. <i>Separation Logic</i> . Contracte în POO. Spec#. SAT și SMT solve. Logică temporală și utilizarea algoritmilor de model-checking. K-Framework și Matching Logic.
---------------------	---

BIBLIOGRAFIE (SELECTIVĂ)	B. Meyer: Object-Oriented Software Construction, second edition, Prentice Hall, January 1997. Michael Huth and Mark Ryan. Logic in Computer Science, modelling and reasoning about systems (2nd edition). Cambridge University Press, 2004. John C. Reynolds. An Introduction to Separation Logic, 2009. A tutorial on Program Verification using the Spec# Programming System, presented at ECOOP 2009. Dorel Lucanu Grigore Rosu Traian Serbanuta . The K Framework Distilled.
--------------------------	--

CONȚINUTUL LUCRĂRILOR DE SEMINAR/LABORATOR	În prima parte se va pune accent pe exerciții de formalizare a problemelor și de aplicare diferitelor sisteme de verificare a programelor. În partea a doua se vor prezenta sub formă de referate și dezbate diferite proiecte realizate la diverse institute de cercetare sau universități.
--	---

BIBLIOGRAFIE (SELECTIVĂ)	B. Meyer: Object-Oriented Software Construction, second edition, Prentice Hall, January 1997. Michael Huth and Mark Ryan. Logic in Computer Science, modelling and reasoning about systems (2nd edition). Cambridge University Press, 2004. John C. Reynolds. An Introduction to Separation Logic, 2009. A tutorial on Program Verification using the Spec# Programming System, presented at ECOOP 2009.
--------------------------	---

	Dorel Lucanu Grigore Rosu Traian Serbanuta . The K Framework Distilled.
REPERE METODOLOGICE***	Predare interactivă cu utilizarea combinată a videoproietorului și a tablei, prezentări și dezbateri de proiecte.

EVALUARE	metode	activitate seminar, proiecte individuale, test scris
	forme	Activitate seminar: punerea de întrebări, participarea la discuții, propunerea de subiecte de discuții, referate. Testul scris: se va da în ultima săptămână din semestru și va include întrebări din toată materia. Prezentarea de referate constituie un bonus sau echivalare parțială a celorlalte activități. Prezentarea a două referate este echivalată cu un test scris.
	ponderea formelor de evaluare în formula notei finale	Activitate la seminar (60%), test scris (40%);
	standardele minime de performanță****	Realizarea unui proiect care în care se aplică metode de verificare pe un set de programe concrete și a unui proiect care să analizeze definiția unui limbaj de programare în K Framework.

* obiectivele sunt formulate în funcție de grila competențelor profesionale pentru programul de studii

** la nivel de descriptor

*** strategia didactică, materiale, resurse

**** raportate la competențele formulate la Obiective sau la Standardele minime de performanță din grila 1L/1M după caz

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar/l.p.

Data avizării în departament

Semnătura directorului de departament