

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai din Cluj-Napoca
1.2 Facultatea	Facultatea de Matematică și Informatică
1.3 Departamentul	Departamentul de Informatică
1.4 Domeniul de studii	Informatica
1.5 Ciclul de studii	Postuniversitar
1.6 Programul de studiu / Calificarea	Program postuniversitar de formare si dezvoltare profesionala in Informatica

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)	Verificare, validare și testare automată						
2.2 Titularul activităților de curs	Prof. dr. Csató Lehel						
2.3 Titularul activităților de seminar	Prof. dr. Csató Lehel						
2.4 Anul de studiu	1	2.5 Semestrul	2	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie
2.8 Codul disciplinei	MLM5131						

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	4	Din care: 3.2 curs	2	3.3 seminar/laborator	2
3.4 Total ore din planul de învățământ	40	Din care: 3.5 curs	20	3.6 seminar/laborator	20
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					30
Tutoriat					6
Examinări					4
Alte activități:					0
3.7 Total ore studiu individual		85			
3.8 Total ore pe semestru		125			
3.9 Numărul de credite		5			

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> Fundamentele Programarii si Algoritmica, Programare si Structuri de Date, Programare Orientata Obiect, Metode și Medii avansate de programare
4.2 de competențe	<ul style="list-style-type: none"> Cunoștințe de programare într-un limbaj de programare de nivel înalt, orientat obiect

5. Condiții (acolo unde este cazul)

5.1 De desfășurare a cursului	<ul style="list-style-type: none"> • Sală, plus proiector
5.2 De desfășurare a seminarului/laboratorului	<ul style="list-style-type: none"> • Sală de laborator cu calculatoare dotate cu limbajul de programare Java

6. Competențele specifice acumulate

Competențe profesionale	<ul style="list-style-type: none"> • Identificarea de metodologii adecvate de dezvoltare a sistemelor software. • Identificarea și explicarea mecanismelor adecvate de specificare a sistemelor software. • Utilizarea metodologiilor, mecanismelor de specificare și a mediilor de dezvoltare pentru realizarea aplicațiilor informatice. • Utilizarea de criterii și metode adecvate pentru evaluarea aplicațiilor informatice. • Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date • Testarea unor aplicații pe baza unor planuri de test
Competențe transversale	<p>CT1 Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională</p> <p>CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională</p>

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Înțelegerea noțiunilor de algoritm parțial și total corect; • Formarea deprinderilor de proiectare a algoritmilor în paralel cu demonstrarea corectitudinii lor; • Cunoașterea metodelor de testare și verificare a sistemelor soft; • Formarea deprinderilor de proiectare a programelor corecte din specificații; • Formarea unui stil modern de programare.
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • Studenții vor ști cum se desfășoară și care sunt pașii unei inspecții, fie a codului sursă fie a specificației din fiecare etapă de dezvoltare a sistemului soft. • Studenții vor ști să prevadă încă din faza de specificare și proiectare crearea unor cazuri de testare care să-i ajute la dezvoltarea unui sistem soft mai robust. • Studenții vor ști să utilizeze instrumentele pentru managementul procesului de testare. • Studenții vor ști să proiecteze cazurile de testare folosind diferite criterii (black-box, white-box).

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. Verificarea și validarea (V&V) sistemelor soft: introducere și noțiuni fundamentale	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație 	

	<ul style="list-style-type: none"> • Exemple • Demonstrație didactică 	
2. V&V: metode statice (inspectarea sistemelor software, code review, code inspection, analiză statică automatizată)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
3. V&V: metode formale (corectitudinea algoritmilor, contribuțiile lui Floyd și Hoare, metode bazate pe modele, Cleanroom)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
4. V&V: metode dinamice (tehnici de testare, categorizarea metodelor)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
5. Testare automatizată: unit testing (JUnit, Hamcrest), izolarea testelor (Mockito)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
6. Testare automatizată: integration testing (Arquillian)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
7. Testare automatizată: UI testing (Selenium)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
8. Testare automatizată: testarea aplicațiilor mobile (framework-uri pt. UI testing)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
9. V&V și CI/CD (Continuous Integration/Deployment), sisteme pentru monitorizarea aplicațiilor	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
10. V&V în contextul QA (Quality Assurance), recapitulare	<ul style="list-style-type: none"> • Expunere interactivă • Explicație 	

	<ul style="list-style-type: none"> • Conversație • Exemple • Demonstrație didactică 	
--	--	--

Bibliografie

1. Frentiu, M., Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010
2. R. S. Pressman, Software engineering: a practinioner's approach, seventh edition, Higher Education, 2010
3. L. Crispin, J. Grecory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009
4. M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008
5. K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
6. J. P. Katoen, Principles of Model Checking, MIT Press, May 2008
7. R. Patton, Software Testing, Sams Publishing, 2005
8. Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004
9. I. Bernstein, Practical software testing, Springer, 2002
10. Balanescu T., Corectitudinea programelor, Editura tehnica, Bucuresti 1995.
11. Morgan, C., Programing from Specifications, Prentice Hall, NewYork, 1990.

8.2 Seminar / laborator	Metode de predare	Observații
1. Exerciții: testare manuală și debugging	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
2. Exerciții: code review & code inspection	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
3. Folosirea sistemelor de analiză statică automatizată (de ex. SonarQube)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
4. Exerciții: proiectarea testelor	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
5. Implementarea unor teste automatizate folosind JUnit+Hamcrest+Mockito	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
6. Implementarea unor teste automatizate folosind Arquillian	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple 	

	<ul style="list-style-type: none"> • Demonstrație didactică 	
7. Implementarea unor teste automatizate folosind Selenium	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
8. Implementarea unor teste automatizate folosind Espresso/Apium	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
9. Crearea unor pipeline-uri CI/CD în contextul metodelor V&V/QA (teste automatizate, instalare automatizată pe servere de testare, rularea automatizată al analizelor statice automatizate)	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	
10. Documentarea testelor, folosirea sistemelor de proiect management/issue tracking în contextul activităților legate de V&V/QA	<ul style="list-style-type: none"> • Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică 	

Bibliografie

1. Frentiu, M., Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010
2. R. S. Pressman, Software engineering: a practinioner's approach, seventh edition, Higher Education, 2010
3. L. Crispin, J. Grecory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009
4. M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008
5. K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
6. J. P. Katoen, Principles of Model Checking, MIT Press, May 2008
7. R. Patton, Software Testing, Sams Publishing, 2005
8. Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004
9. I. Bernstein, Practical software testing, Springer, 2002
10. Balanescu T., Corectitudinea programelor, Editura tehnica, Bucuresti 1995.
11. Morgan, C., Programing from Specifications, Prentice Hall, NewYork, 1990.

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

- Cursul respecta recomandările IEEE și ACM legate de Curiculla pentru specializarea Informatică.
- Cursul face parte din programul de studiu de la majoritatea universităților importante din România și din străinătate.
- Conținutul cursului este considerat de companiile soft ca fiind important pentru un nivel mediu de cunoștințe în programare.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 metode de evaluare	10.3 Pondere din nota finală
----------------	---------------------------	-------------------------	------------------------------

10.4 Curs	Corectitudinea și completitudinea cunoștințelor acumulate.	Examen scris	50%
10.5 Seminar/laborator	Abilitatea de a proiecta și efectua teste, de a crea teste automatizate	Examen practic	50%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • Minimum 5 la fiecare proba. 			

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

Prof. dr. Csató Lehel

Prof. dr. Csató Lehel

Data avizării în departament

Semnătura directorului de departament

Conf. dr. András Szilárd Károly