FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	**Universitatea Babeş-Bolyai din Cluj-Napoca**
1.2 Facultatea	Facultatea de Matematică şi Infromatică
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 informatică și dezvoltare software (în limba maghiară)**

2. Date despre disciplină

2.1 Denumirea disciplinei (ro) (en)	**Verificare, validare și testare automată**						
2.2 Titularul activităţilor de curs		Conf. dr. Gaskó Noémi					
2.3 Titularul activităţilor de seminar		Conf. dr. Gaskó Noémi					
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 activităţilor didactice)

3.1 Număr de ore pe săptămână	5	Din care: 3.2 curs	2	3.3 seminar/laborator	3
3.4 Total ore din planul de învăţământ	50	Din care: 3.5 curs	20	3.6 seminar/laborator	30
Distribuţia fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie şi notiţe					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren					15
Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri					30
Tutoriat					6
Examinări					4
Alte activităţi:					0

3.7 Total ore studiu individual	75
3.8 Total ore pe semestru	125
3.9 Numărul de credite	5

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	• Fundamentele Programarii si Algoritmica, Programare si Structuri de Date, Programare Orientata Obiect, Metode și Medii avansate de programare
4.2 de competenţe	• Cunoștințe de programare într-un limbaj de programare de nivel înalt, orientat obiect

5. Condiţii (acolo unde este cazul)

5.1 De desfăşurare a cursului	• Sală, plus proiector
5.2 De desfăşurare a seminarului/laboratorului	• Sală de laborator cu calculatoare dotate cu limbajul de programare Java

6. Competenţele specifice acumulate

Competenţe profesionale	• Identificarea de metodologii adecvate de dezvoltare a sistemelor software. • Identificarea şi 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 şi testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificaţii de proiectare date • Testarea unor aplicații pe baza unor planuri de test
Competenţe transversale	CT1 Aplicarea regulilor de muncă organizată şi eficientă, a unor atitudini responsabile faţă de domeniul didactic-ştiinţific, pentru valorificarea creativă a propriului potenţial, cu respectarea principiilor şi a normelor de etică profesională CT3 Utilizarea unor metode şi tehnici eficiente de învăţare, informare, cercetare şi dezvoltare a capacităţilor de valorificare a cunoştinţelor, de adaptare la cerinţele unei societăţi dinamice și de comunicare în limba română și într-o limbă de circulație internațională

7. Obiectivele disciplinei (reieşind din grila competenţelor acumulate)

7.1 Obiectivul general al disciplinei	• Intelegerea notiunilor de algoritm partial si total corect; • Formarea deprinderilor de proiectare a algoritmilor in paralel cu demonstrarea corectitudinii lor; • Cunoasterea metodelor de testare si verificare a sistemelor soft; • Formarea deprinderilor de proiectare a programelor corecte din specificatii; • Formarea unui stil modern de programare.
7.2 Obiectivele specifice	• Studentii vor sti cum se desfasoara si care sunt pasii unei inspectari, fie a codului sursa fie a specificatie din fiecare etapa de dezvoltare a sistemului soft. • Studentii vor sti sa prevada inca din faza de specificare si proiectare crearea unor cazuri de testare care sa-i ajute la dezvoltarea unui sistem soft mai robust. • Studenţii vor sti să utilizeze instrumentele pentru managementul procesului de testare. • Studenţii vor şti să proiecteze cazurile de testare folosind diferite criterii (black-box, white-box).

8. Conţinuturi

8.1 Curs	Metode de predare	Observaţii
1. Verificarea si validarea (V&V) sistemelor soft: introducere și noțiuni fundamentale	• Expunere interactivă • Explicație • Conversație	

	• Exemple • Demonstrație didactică	
2. V&V: metode statice (inspectarea sistemelor software, code review, code inspection, analiză statică automatizată)	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
3. V&V: metode formale (corectitudinea algoritmilor, contributiile lui Floyd și Hoare, metode bazate pe modele, Cleanroom)	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
4. V&V: metode dinamice (tehnici de testare, categorizarea metodelor)	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
5. Testare automatizată: unit testing (JUnit, Hamcrest), izolarea testelor (Mockito)	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
6. Testare automatizată: integration testing (Arquillian)	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
7. Testare automatizată: UI testing (Selenium)	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
8. Testare automatizată: testarea aplicațiilor mobile (framework-uri pt. UI testing)	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
9. V&V și CI/CD (Continuous Integration/Deployment), sisteme pentru monitorizarea aplicațiilor	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
10. V&V în contextul QA (Quality Assurance), recapitulare	• 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.		
8.2 Seminar / laborator	Metode de predare	Observaţii
1. Exerciții: testare manuală și debugging	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
2. Exerciții: code review & code inspection	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
3. Folosirea sistemelor de analiză statică automatizată (de ex. SonarQube)	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
4. Exerciții: proiectarea testelor	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
5. Implementarea unor teste automatizate folosind JUnit+Hamcrest+Mockito	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
6. Implementarea unor teste automatizate folosind Arquillian	• Expunere interactivă • Explicație • Conversație • Exemple	

	• Demonstrație didactică	
7. Implementarea unor teste automatizate folosind Selenium	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
8. Implementarea unor teste automatizate folosind Espresso/Apium	• 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)	• Expunere interactivă • Explicație • Conversație • Exemple • Demonstrație didactică	
10. Documentarea testelor, folosirea sistemelor de project management/issue tracking în contextul activităților legate de V&V/QA	• 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 conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatori reprezentativi din domeniul aferent programului

- Cursul respecta recomandările IEEE şi 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 performanţă
• Minimum 5 la fiecare proba.

Data completării	Semnătura titularului de curs	Semnătura titularului de seminar
30.08.2020	Conf. dr. Gaskó Noémi	Conf. dr. Gaskó Noémi

Data avizării în departament

Semnătura directorului de departament

Conf. dr. András Szilárd Károly