FIŞA DISCIPLINEI

Verificarea si Validarea Sistemelor Software

Anul universitar 2025-2026

1. Date despre program

1.1. Instituţia de învăţământ superior	Universitatea Babes-Bolyai
1.2. Facultatea	Facultatea de Matematica si Informatica
1.3. Departamentul	Departamentul de Informatica
1.4. Domeniul de studii	Informatica
1.5. Ciclul de studii	Licenta
1.6. Programul de studii / Calificarea	Inteligenta Artificiala
1.7. Forma de învăţământ	Cu frecventa

2. Date despre disciplină

2.1. Denumirea disciplinei	**Verificarea si Validarea Sistemelor Software**					Codul disciplinei	**MLE5014**
2.2. Titularul activităţilor de curs	Conf. Dr. Vescan Andreea						
2.3. Titularul activităţilor de seminar	Conf. Dr. Vescan Andreea						
2.4. Anul de studiu	3	2.5. Semestrul	6	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Optional

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

3.1. Număr de ore pe săptămână	**3**	din care: 3.2. curs	**2**	3.3. seminar/ laborator/proiect	**1**
3.4. Total ore din planul de învăţământ	36	din care: 3.5. curs	24	3.6 seminar/laborator/proiect	**12**
Distribuţia fondului de timp pentru studiul individual (SI) şi activităţi de autoinstruire (AI)					**Ore**
Studiul după manual, suport de curs, bibliografie şi notiţe (AI)					24
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren					24
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii şi eseuri					24
Tutoriat (consiliere profesională)					7
Examinări					10
Alte activităţi					0
3.7. Total ore studiu individual (SI) şi activităţi de autoinstruire (AI)				**89**	
3.8. Total ore pe semestru				**125**	
3.9. Numărul de credite				**5**	

4. Precondiţii (acolo unde este cazul)

4.1. de curriculum	Programare orientată pe obiecte, Metode avansate de programare, Sisteme de proiectare şi implementare, Programare Web
4.2. de competenţe	Abilităţi în medii de programare la nivel înalt orientat obiect

5. Condiţii (acolo unde este cazul)

5.1. de desfăşurare a cursului	Video proiector, acces la internet
5.2. de desfăşurare a seminarului/ laboratorului	Laborator cu calculatoare; diverse instrumente pentru activităţi de testare/verificare

6.1. Competenţele specifice acumulate[1]

[1] Se poate opta pentru competenţe sau pentru rezultatele învăţării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va şterge tabelul aferent celeilalte opţiuni, iar opţiunea păstrată va fi numerotată cu 6.

Competenţe profesionale/esenţiale	• programarea în limbaje de nivel înalt • dezvoltarea şi întreţinerea aplicaţiilor informatice • utilizarea instrumentelor informatice în context interdisciplinar
Competenţe transversale	• desfăşurarea eficientă a activităţilor organizate într-un grup interdisciplinar şi dezvoltarea capacităţilor empatice de comunicare interpersonală, de relaţionare şi colaborare cu grupuri diverse • 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 şi de comunicare în limba română şi într-o limbă de circulaţie internaţională

6.2. Rezultatele învăţării

Cunoştinţe	Studentul cunoaşte: • Absolventul are cunoştinţe adecvate legate de folosirea mediilor de dezvoltare integrate in scopul creării de aplicaţii complexe de dimensiuni mari. • Absolventul este familiarizat cu metodologiile de dezvoltare tradiţionale şi agile.
Aptitudini	Studentul este capabil să: • Absolventul are abilitatea de a crea teste automate de diferite nivele de granularitate pentru asigurarea calităţii sistemelor dezvoltate. • Absolventul este familiar cu instrumente folosite pentru testarea, depanarea, validarea aplicaţii software. • Absolventul cunoaşte metodele de testare şi verificare a sistemelor software. • Absolventul este familiarizat cu instrumentele de management de proiect, sistemele de control al versiunilor, precum şi conceptele, metodele, instrumentele de continuous integration/continuous delivery (CI/CD).
Responsabilităţi şi autonomie	Studentul are capacitatea de a lucra independent pentru: • Absolventul are abilitatea de a înţelege şi comunica eficient informaţiile. • Absolventul are capacitatea de a observa şi obţine informaţii din diverse surse.

7. Obiectivele disciplinei (reieşind din grila competenţelor acumulate)

7.1 Obiectivul general al disciplinei	• Să dobândească cunoştinţe despre algoritmii parţial corecţi şi total corecţi • Să dobândească cunoştinţe despre proiectarea algoritmilor corecţi şi demonstrarea corectitudinii acestora; • Să înveţe metodele de verificare şi validare a programelor; • Să se obişnuiască cu construirea de programe corecte din specificaţii; • Să dezvolte un stil modern de programare.

7.2 Obiectivele specifice	• Studenţii vor şti cum şi care sunt paşii unei inspecţii, fie a codului sursă, fie a specificaţiei fiecărei etape a dezvoltării sistemului software. • Studenţii vor şti să creeze cazuri de testare din specificaţie şi din codul sursă, ceea ce îi va ajuta să dezvolte un sistem software mai bun şi robust. • Studenţii vor şti cum să utilizeze instrumente pentru gestionarea procesului de testare. • Studenţii vor şti cum să proiecteze cazuri de testare folosind diverse criterii (black-box, white-box).

8. Conţinuturi

8.1 Curs	Metode de predare	Observaţii
Verificare si validare. Inspectarea programelor	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Testarea programelor (1): conceptul de testare a programelor; testare unitară: criterii de testare – testare cutie neagră,	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Testare programelor (2): conceptul de testare program; testare unitară: criterii de testare – testare cutie albă (continuare)	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Testarea programelor (3): Niveluri de testare (unitate, integrare, sistem, regresie, acceptare)	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Testarea aplicațiilor web	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Testare agilă. Testarea scripturilor versus testarea exploratorie	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Symbolic execution	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Model checking	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Teoria corectitudinii programului. Evoluția conceptului de corectitudine a programului. --- Metoda lui Floyd pentru a demonstra corectitudinea. --- Metoda de axiomatizare a lui Hoare --- Dijkstra: cea mai slabă condiție prealabilă. Rafinament în trepte din specificații	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Calitatea programului	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	

Activități legate de verificare/testare: abilități de testare tehnică, abilități de testare soft, feedback. Această activitate se realizează în colaborare a profesorului cu studentii.	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	
Pregătirea examenului final.	Expunere interactivă Explicaţie Conversaţie Demonstratie didactica	

Bibliografie

Bibliography

Books

[Fre10] FRENTIU, M., Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010
[Pres10] R. S. Pressman, Software engineering: a practinioner's approach, seventh edition, Higher Education, 2010
[Crs09] L. Crispin, J. Grecory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009
[You08] M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008
[Nai08] K. Naik, P. Tripathy, Software testing and quality assurance. Theory and Practice, A John Wiley & Sons, Inc., 2008
[Kat08] J. P. Katoen, Principles of Model Checking, MIT Press, May 2008
[Pat05] R. Patton, Software Testing, Sams Publishing, 2005
[Mye04] Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, Inc., 2004
[Brn02] I. Brnstein, Practical software testing, Springer, 2002
[Mor90] Morgan, C., Programing from Specifications, Prentice Hall, NewYork, 1990.
[Dro89] DROMEY G., Program Derivation. The Development of Programs From Specifications, Addison Wesley Publishing Company, 1989.

Articles

[Kin75] J. Darringer, J. King, Applications of symbolic execution to program testing, 1975
[Dij75] DIJKSTRA, E., Guarded commands, nondeterminacy and formal derivation of programs, CACM, 18(1975), 8, pg.453-457.
[Hoa69] HOARE, C.A.R., An axiomatic basis for computer programming, CACM, 12(1969), pg.576-580, 583.

Tutorials

În timpul prelegerilor/seminarelor/laboratoarelor vor fi oferite tutoriale pentru fiecare temă.

8.2 Seminar / laborator	Metode de predare	Observaţii
Seminar 1/Laborator 1 Inspectare Instrument de inspectare	Prezentare, Conversaţie, Problematizări, Descoperire, Alte metode – studiu individual, exerciţii	
Seminar 2/Laborator 2 Cazuri de testare folosind Black-box Testing (BBT) Instrument de integrare continuă (Jenkins)	Prezentare, Conversaţie, Problematizări, Descoperire, Alte metode – studiu individual, exerciţii	
Seminar 3/Laborator 3 Cazuri de testare folosind White-box Testing (WBT) Instrument de integrare continuă (Jenkins)	Prezentare, Conversaţie, Problematizări, Descoperire, Alte metode – studiu individual, exerciţii	
Seminar 4/Laborator 4	Prezentare, Conversaţie,	

Niveluri de testare - Testare de integrare Instrument de integrare continuă (Jenkins)	Problematizări, Descoperire, Alte metode – studiu individual, exerciţii	
Seminar 5/Laborator 5 Testare web Instrument de testare web (de exemplu, Selenium Web Driver) Instrument de integrare continuă (Jenkins)	Prezentare, Conversaţie, Problematizări, Descoperire, Alte metode – studiu individual, exerciţii	
Seminar 6/Laborator 6 Corectitudine. Analiza statica ESCJava2, JML	Prezentare, Conversaţie, Problematizări, Descoperire, Alte metode – studiu individual, exerciţii	
Bibliografie Vedeţi referinţele din Prelegeri. Remarcă. Pentru fiecare seminar, studenţii trebuie să fie pregătiţi. Diverse articole/capitole din cărţi trebuie citite înainte de fiecare seminar.		

9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatori reprezentativi din domeniul aferent programului

- Studentii vor şti cum să folosească instrumente pentru managementul testelor
- Studentii vor şti cum să aplice metode de testare pentru un produs software.
- Studenţii vor învăţa diverse metode de verificare şi validare a unui sistem software, pentru a proiecta cazuri de testare folosind diverse criterii (testare cutie neagră, testare cutie albă)

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.1 Curs	La sfârşitul semestrului un scris examenul va da Nota E.	Examen Scris	50%
10.2 Seminar/laborator	Activitatea de la seminarii, constând din participare la rezolvarea exercitiilor iar discuţiile vor fi notate cu nota S.	Seminar = Nota pentru Activitatea seminarului	25%
	Activitatea de la laboratoare, constând din participare la rezolvarea exercitiilor şi discuţii, vor fi notate cu nota L.	Laborator activitate	25%
10.3. Puncte bonus	Studenţii vor avea posibilitatea de a obţine puncte bonus la nota finală pentru activităţi suplimentare care sunt legate de verificarea şi validarea sistemelor software: cercetare/raport de conducere şi diverse activităţi în timpul prelegerilor. De asemenea, ar putea fi selectat un proiect de cercetare şi dezvoltare.	Puncte bonus	Puncte bonus la nota finală (după obţinerea minimului final de nota 5).

Observaţie. • Seminarul/Temele de laborator/Temele practice de laborator nu pot fi refăcute în sesiunea de reluare. • Examenele scrise pot fi susţinute în timpul sesiunii de reluare. • Studenţi din anii anteriori până în anul universitar curent o Toate regulile de mai sus se aplică studenţilor din anii anteriori. o Temele de seminar/laborator şi activitatea practică de laborator trebuie refăcute în timpul activităţii didactice (în cele 12 săptămâni înainte de sesiunea normală). • Activitate de laborator: fiecare student va veni cu propria semigrupă. • Activitate de laborator: 3 din 6 laboratoare trebuie predate. • Predare cu întârziere a sarcinilor va fi penalizată. Sunt permise maxim 4 săptămâni pentru a preda o tema. După termenul limită, tema va fi notată cu 0. • Nota finală calculată cu formula dată trebuie să fie minim 5 pentru a promova examenul. Nota finala=50%Examen scris+25%Seminar+25%Laborator • Participaţi la 75% din activităţile de seminar în timpul semestrului ŞI participaţi la 90% din activităţile de laborator în timpul semestrului.
10.6 Standard minim de performanţă
•

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)[2]

Nu se aplică.

Data completării: ...	Semnătura titularului de curs Conf. Dr. Vescan Andreea	Semnătura titularului de seminar Conf. Dr. Vescan Andreea
Data avizării în departament: ...		Semnătura directorului de departament Conf.dr. Adrian STERCA

[2] Păstraţi doar etichetele care, în conformitate cu *Procedura de aplicare a etichetelor ODD în procesul academic*, se potrivesc disciplinei şi ştergeţi-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ştergeţi-le pe toate şi scrieţi "*Nu se aplică.*".