

FIȘA DISCIPLINEI

Verificarea și Validarea Sistemelor Soft

Anul universitar 2025 – 2026

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai
1.2. Facultatea	Facultatea de Matematică și Informatică
1.3. Departamentul	Departamentul de Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Informatică (în limba germană)
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei		Verificarea și Validarea Sistemelor Soft (Software Systems Verification and Validation)				Codul disciplinei	MLG5014
2.2. Titularul activităților de curs		Lector dr. ing. Kuderna-Iulian Bența					
2.3. Titularul activităților de seminar		Lector dr. ing. Kuderna-Iulian Bența					
2.4. Anul de studiu	3	2.5. Semestrul	6	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligatorie

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/proiect	2
3.4. Total ore din planul de învățământ	48	din care: 3.5. curs	24	3.6 seminar/laborator/proiect	24
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					34
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					35
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					46
Tutoriat (consiliere profesională)					6
Examinări					6
Alte activități					0
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				127	
3.8. Total ore pe semestru				175	
3.9. Numărul de credite				7	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Programare orientată obiect, Metode avansate de programare, Medii de proiectare și programare, Programare Web
4.2. de competențe	Cunoașterea mediilor de programare pentru dezvoltarea aplicațiilor în limbaje orientate-obiect de nivel înalt

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs cu videoproiector
5.2. de desfășurare a seminarului/ laboratorului	Laborator

6.1. Competențele specifice acumulate¹

¹ Se poate opta pentru competențe sau pentru rezultatele învățării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va șterge tabelul aferent celeilalte opțiuni, iar opțiunea păstrată va fi numerotată cu 6.

Competențe profesionale/esențiale	• programarea în limbaje de nivel înalt • dezvoltarea și întreținerea aplicațiilor informatice • utilizarea instrumentelor informatice în context interdisciplinar • utilizarea bazelor teoretice ale informaticii și a modelelor formale
Competențe transversale	• 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 • desfășurarea eficientă a activităților organizate într-un grup interdisciplinar și dezvoltarea capacităților empatice de comunicare interpersonală, de relaționare și colaborare cu grupuri diverse

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște metodele de testare și verificare a sistemelor software. Studentul este familiarizat cu instrumente folosite pentru testarea, depanarea, validarea aplicații software. Studentul este familiarizat cu instrumentele de management de proiect, sistemele de control al versiunilor, precum și conceptele, metodele, instrumentele de continuous integration/continuous delivery (CI/CD).
Aptitudini	Studentul are abilitatea de a alege și folosi module și medii existente pentru dezvoltarea de aplicațiilor. Studentul este capabil să aplice șabloane arhitecturale, șabloane de proiectare și bunele practici în domeniu pentru a proiecta aplicații software de complexitate mare. Studentul are abilitatea de a crea teste automate de diferite nivele de granularitate pentru asigurarea calității sistemelor dezvoltate.
Responsabilități și autonomie	Studentul are capacitatea de a observa și obține informații din diverse surse. Studentul are capacitatea de a identifica probleme complexe și să examineze probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții. Studentul are capacitatea de a evalua diferite arhitecturi și soluții posibile pentru o problemă și a alege pe cele potrivite pentru cerințele și constrângerile specifice aplicației de dezvoltat.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	• 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	• Studenții vor ști cum se desfășoară și care sunt pașii unei inspectări, 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 sistemelor soft. Inspectarea programelor	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
2. Testarea programelor (1): Conceptul de testare a programelor. Criterii de testare. Testarea black-box	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
3. Testarea programelor (2): Criterii de testare. Testarea white-box	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
4. Niveluri de testare. Tipuri de testare	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
5. Testarea aplicațiilor Web. SeleniumWebDriver	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
6. Execuție simbolică	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
7. Verificarea modelelor	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
8. Teoria corectitudinii programelor (I) • Evoluția conceptului de corectitudine • Contribuțiile lui Floyd, Hoare	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
9. Teoria corectitudinii programelor (II) • Contribuțiile lui Dijkstra	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
10. Asigurarea calității produselor soft. Controlul calității	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
11. Abilități în testare și atribuții ale testerului	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
12. Susținerea referatelor	Expunere interactivă, Explicații, Conversație, Demonstrație didactică	
Bibliografie: 1. Kleuker, S., Formale Modelle der Softwareentwicklung, Vieweg Teubner, 2009. 2. Haubelt, C., Teich, J., Digitale Hardware/Software-Systeme, Spezifikation und Verifikation, Springer, 2010. 3. A. Spillner, M. Winter, A. Pietschker (Hrsg.), Test, Analyse und Verifikation von Software – gestern, heute, morgen, November 2017, 224 Seiten, Broschur, dpunkt.verlag, ISBN Print: 978-3-86490-470-7 4. R. Oshero, The Art of Unit Testing 2. Auflage 2015, 312 Seiten, MITP, ISBN: 9783826697128 Bibliografie în alte limbi: 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.		
8.2 Seminar / laborator	Metode de predare	Observații
S1: Inspectare L1: Inspectare Unelte de inspectare. Unelte Issue traker. Unelte de gestiune a testării	Proiect, Prezentare, Conversație, Problematizare, Descoperire, Studiu individual, Exerciții.	
S2: Specificarea cazurilor de testare pe baza specificațiilor (BBT) L2: Proiectarea cazurilor de testare pe baza specificațiilor (BBT) Instrumente de gestiune a testării. Instrumente pentru integrare continuă	Proiect, Prezentare, Conversație, Problematizare, Descoperire, Studiu individual, Exerciții.	
S3: Specificarea cazurilor de testare pe baza codului sursa (WBT) L3: Proiectarea cazurilor de testare pe baza codului sursa (WBT) Instrumente de gestiune a testării. Instrumente pentru integrare continuă	Proiect, Prezentare, Conversație, Problematizare, Descoperire, Studiu individual, Exerciții.	

S4: Niveluri de testare L4: Niveluri de testare Instrumente de gestiune a testării. Instrumente pentru integrare continuă	Proiect, Prezentare, Conversație, Problematizare, Descoperire, Studiu individual, Exerciții.	
S5: Corectitudine. Floyd L5: TestareGUI/Web Instrumente pentru testare Web. Instrumente de gestiune a testării. Unelte pentru integrare continuă	Proiect, Prezentare, Conversație, Problematizare, Descoperire, Studiu individual, Exerciții.	
S6: Rafinare din specificații L6: Analiză statică: JML, ESC2Java	Proiect, Prezentare, Conversație, Problematizare, Descoperire, Studiu individual, Exerciții.	
Bibliografie: 5. Kleuker, S., Formale Modelle der Softwareentwicklung, Vieweg Teubner, 2009. 6. Haubelt, C., Teich, J., Digitale Hardware/Software-Systeme, Spezifikation und Verifikation, Springer, 2010. 7. A. Spillner, M. Winter, A. Pietschker (Hrsg.), Test, Analyse und Verifikation von Software – gestern, heute, morgen, November 2017, 224 Seiten, Broschur, dpunkt.verlag, ISBN Print: 978-3-86490-470-7 8. R. Oshero, The Art of Unit Testing 2. Auflage 2015, 312 Seiten, MITP, ISBN: 9783826697128 Bibliografie în alte limbi: 5. Frentiu, M., Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010 6. R. S. Pressman, Software engineering: a practinioner's approach, seventh edition, Higher Education, 2010 7. L. Crispin, J. Greco, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009 8. M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008.		

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

- Studenții vor dobândi abilitatea de a utiliza instrumentele pentru gestiunea procesului de testare - Studenților li se vor prezenta diferite metode de testare pe care le vor aplica produselor soft Studenții vor dobândi abilitatea de a proiecta cazurile de testare folosind diferite criterii (black-box, white-box)

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Examinarea scrisă (nota E)	Examen scris în sesiune	40%
	Prezentarea referatelor pe teme de verificare și validare (nota R)	Evaluare orală	Bonus (la nota finală, între 0 și 2 puncte)
10.5 Seminar + laborator	Participarea la discuții și la rezolvarea unor exerciții va fi notată (nota S)	Activitate la Seminar	10%
	Activitatea de laborator va fi notată (nota L) pe baza cerințelor	Evaluarea temelor de laborator	50%
Observații: - Activitatea de laborator nu se va re-evalua în timpul sesiunilor de examinare sau de re-examinare. - Activitatea de laborator pentru studenții restanțieri trebuie refăcută. - Prezența la activitatea de laborator se poate face doar cu grupa din care studentul face parte. - Predarea cu întârziere se va penaliza cu 2 puncte din nota acordată. După depășirea termenului de predare, tema de laborator va fi notată cu 1.			
10.6 Standard minim de performanță			
• Abilitatea practică de a utiliza instrumentele pentru managementul procesului de testare • Înțelegerea teoretică și abilitatea practică de a utiliza diferite criterii de proiectare a cazurilor de testare (black-box, white-box). • Cunoașterea diferitelor metode de verificare (inspectare, testare, demonstrarea corectitudinii).			

- Condițiile de participare la examenul final: 75% din activitățile de seminar (minimum 4 prezențe) și 90% din activitățile de laborator (minimum 5 prezențe).

Promovarea disciplinei presupune participarea la examenul din sesiune și obținerea mediei finale $M \geq 5.00$, unde $M = 40\%E + 10\%S + 50\%L + R$, conform explicațiilor de mai sus.

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

Nu se aplică.

Data completării:
Aprilie 2025

Semnătura titularului de curs
Lector Dr. ing. Kuderna-Iulian Bența

Semnătura titularului de seminar
Lector Dr. ing. Kuderna-Iulian Bența

Data avizării în departament:
Aprilie 2025

Semnătura directorului de departament
Conf.dr. Adrian STERCA

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