

FIȘA DISCIPLINEI

Elaborarea lucrării de licență

Anul universitar 2026-2027

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Babeș-Bolyai
1.2. Facultatea	Facultatea de Matematica și Informatică
1.3. Departamentul	Departamentul de Informatică
1.4. Domeniul de studii	Calculatoare și tehnologia informației
1.5. Ciclul de studii	Licență
1.6. Programul de studii / Calificarea	Ingineria informației
1.7. Forma de învățământ	Cu frecvență

2. Date despre disciplină

2.1. Denumirea disciplinei	Inginerie software			Codul disciplinei	MLE5177
2.2. Titularul activităților de curs	Conf. Dr. Vescan Andreea				
2.3. Titularul activităților de seminar	Conf. Dr. Vescan Andreea				
2.4. Anul de studiu	3	2.5. Semestrul	6	2.6. Tipul de evaluare	Examen
2.7. Regimul disciplinei	Obligativ		2.8. Tipul disciplinei	Disciplină de specializare (DS)	

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

3.1. Număr de ore pe săptămână	5	din care: 3.2. curs	2	3.3. seminar/ laborator/ proiect	1S+ 1LP +1P
3.4. Total ore din planul de învățământ	70	din care: 3.5. curs	28	3.6 seminar/laborator	42
Distribuția fondului de timp pentru studiul individual (SI) și activități de autoinstruire (AI)					ore
Studiul după manual, suport de curs, bibliografie și notițe (AI)					20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					20
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii și eseuri					20
Tutoriat (consiliere profesională)					10
Examinări					10
Alte activități					0
3.7. Total ore studiu individual (SI) și activități de autoinstruire (AI)				80	
3.8. Total ore pe semestru				150	
3.9. Numărul de credite				6	

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	Fundamentele programării Programare orientată obiect
4.2. de competențe	Abilități de programare într-un limbaj orienta-obiect de nivel înalt

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Videoproiector
5.2. de desfășurare a seminarului/ laboratorului	<ul style="list-style-type: none">• Calculatoare• Instrument CASE UML• Java/.NET IDE

6.1. Competențele dobândite în urma absolvirii programului de studii (se preiau din planul de învățământ)¹

Competențe profesionale	
Codul competenței	Competență
CP1	crează softuri
CP3	analizează specificații software
CP9	remediază erorile din software
Competențe transversale	
Codul competenței	Competență
CT1	Lucrează independent
CT3	Gândește analitic

6.2. Rezultatele învățării specifice programului de studii (se preiau din planul de învățământ)²

Rezultatele învățării vizate prin disciplină		
Codul competenței	Cunoștințe și înțelegere (Knowledge and understanding)	Abilități academice specifice (Specific academic skills)
CP1	1. Studentul/absolventul identifică, explică și argumentează concepte fundamentale de structuri de date, algoritmi și paradigme de programare, precum și a arhitecturii calculatoarelor.	1. Studentul/absolventul elaborează, dezvoltă și demonstrează soluții software complexe utilizând algoritmi eficienți și paradigme diverse de programare.
CP3	2. Studentul/absolventul numește, oferă exemple, concluzionează, specifică, recunoaște și argumentează critic metodele de proiectare și management al proiectelor informatice complexe, utilizând strategii moderne.	2. Studentul/absolventul inițiază, pregătește, realizează, propune metode de dezvoltare a proiectelor informatice complexe. Studentul/absolventul realizează rapoarte profesionale specifice.
CP9	3. Studentul/absolventul alege, descrie, analizează și explică paradigmele moderne de programare, inclusiv programarea funcțională, orientată pe obiect și paralelă, utilizând limbaje și framework-uri actuale.	3. Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software scalabile și utilizează eficient resursele hardware și software.
CT1	Studentul/absolventul are cunoștințele necesare pentru a înțelege și soluționa probleme complexe, pentru a planifica și organiza procese avansate în diverse domenii.	Absolventul este capabil să identifice probleme complexe și să examineze probleme conexe pentru a dezvolta opțiuni de rezolvare și implementa soluții.
CT3	Studentul/absolventul are cunoștințele necesare pentru a înțelege și soluționa probleme complexe, pentru a planifica și organiza procese avansate în diverse domenii.	Absolventul are abilitatea de a aplica reguli generale unor probleme specifice și de a produce soluții relevante.

7. Rezultatele învățării specifice disciplinei

Cunoștințe și înțelegere (Knowledge and understanding)
1. Absolventul are abilitatea de a înțelege și comunica eficient informațiile.

¹ Se vor prelua din Planul de învățământ al programului de studii acele competențe profesionale și/sau transversale la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa disciplinei. Pentru fiecare competență se va prelua întregul enunț, inclusiv codul competenței, cu formularea care apare în planul de învățământ, fără modificări. Dacă nu se preia nici o competență din oricare din cele două categorii, se șterge linia din tabel aferentă acelei categorii.

² Se menționează rezultatele învățării specifice programului de studiu la dezvoltarea cărora contribuie disciplina pentru care se elaborează fișa. Enunțurile, preluate fără modificări din Planul de învățământ în funcție de tipul disciplinei (DF/DS/DC) se trec în dreptul competenței asociate.

2. Absolventul are capacitatea de a observa și obține informații din diverse surse.
Abilități academice specifice (Specific academic skills)
1. Absolventul este capabil să proiecteze/implementeze componente hardware, software și de comunicații folosind metode de proiectare, limbaje, algoritmi, structuri de date, protocoale și tehnologii, și să le evalueze caracteristicile funcționale și nefuncționale pe baza unor metrici.
2. Absolventul este capabil să dezvolte sisteme și aplicații pentru întreținerea și utilizarea de sisteme hardware, software și de comunicații
3. Absolventul este capabil să realizeze testarea și evaluarea calitativă a caracteristicilor funcționale și nefuncționale ale sistemelor informatice, pe baza unor criterii specifice.
4. Absolventul are abilitatea de a dezvolta, proiecta și crea noi aplicații, sisteme sau produse folosind bunele practici din domeniu.

8. Conținuturi

8.1 Curs	Metode de predare - învățare	Observații ³
Introducere în ingineria sistemelor soft: motivație, definiție, concepte, activități	expunerea, conversația, discutarea unor studii de caz	
2. Ciclul de viață al softului. Modele de procese soft (secvențiale, iterative, specializate)	expunerea, conversația, discutarea unor studii de caz	
3. Tehnici de gestionare a complexității softului (abstractizare, descompunere, modelare). Modelarea în ingineria sistemelor soft: definiții, tipuri de modele și instrumente de modelare	expunerea, conversația, discutarea unor studii de caz	
4. Modelarea sistemelor soft folosind UML: concepte de modelare, tipuri de diagrame, notații, instrumente	expunerea, conversația, discutarea unor studii de caz	
5. Colectarea cerințelor: concepte, activități, exemple	expunerea, conversația, discutarea unor studii de caz	
6. Analiza cerințelor: concepte, activități, exemple	expunerea, conversația, discutarea unor studii de caz	
7. Proiectarea sistemelor soft: concepte, principii, activități. Proiectare arhitecturii	expunerea, conversația, discutarea unor studii de caz	
8. Proiectare sistemelor soft. Proiectare detaliată	expunerea, conversația, discutarea unor studii de caz	
9. Proiectare sistemelor soft. Proiectare detaliată - șabloane de proiectare	expunerea, conversația, discutarea unor studii de caz	

³ De exemplu aspecte organizatorice, recomandări pentru studenți, aspecte specifice legate de curs/seminar cum ar fi invitarea unor practicieni în domeniu etc.

10. Proiectare sistemelor soft. Proiectarea detaliată - specificarea interfețelor	expunerea, conversația, discutarea unor studii de caz	
11. Implementarea sistemelor soft. Transformarea modelelor UML în cod: concepte, principii, activități, exemple	expunerea, conversația, discutarea unor studii de caz	
12. Verificarea și validarea sistemelor soft	expunerea, conversația, discutarea unor studii de caz expunerea, conversația, discutarea unor studii de caz	
13. Gestiunea proiectelor soft: concepte și activități	expunerea, conversația, discutarea unor studii de caz	
14. Metodologii de dezvoltare a sistemelor soft. MDE (Model Driven Engineering)	expunerea, conversația, discutarea unor studii de caz	

Bibliografie

- [1] Booch, G., Rumbaugh, J., Jacobson, I., *The Unified Modeling Language User Guide - V.2.0*, Addison Wesley, 2005.
- [2] Bruegge, B., Dutoit, A., *Object-Oriented Software Engineering Using UML, Patterns and Java - 3rd Edition*, Prentice Hall, 2009.
- [3] Fowler, M. et al., *Refactoring - Improving the Design of Existing Code*, Addison Wesley, 1999.
- [4] Fowler, M., Scott, K., *UML Distilled: A Brief Guide to the Standard Object Modeling Language -2nd ed.*, Addison-Wesley, 1999.
- [5] Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns*, Addison-Wesley, 1996.
- [6] Martin, R.C., *Agile Software Development: Principles, Patterns, and Practices*, Prentice Hall, 2002.
- [7] Pârv, B., *Analiza și proiectarea sistemelor*, Univ. Babeș-Bolyai, CFCID, Facultatea de Matematică și Informatică, Cluj-Napoca, 2004.
- [8] Pressman, R.S., *Software Engineering - A Practitioners Approach - 6th ed.*, McGraw-Hill, 2005.
- [9] Schach, S.R., *Object-Oriented and Classical Software Engineering - 6th ed.*, McGraw-Hill, 2005.
- [10] Sommerville, I., *Software Engineering - 8th edition*, Addison-Wesley, 2006.

Bibliography

8.2 Seminar / laborator	Metode de predare - învățare	Observații
Descrierea modelului funcțional utilizând diagrame de cazuri de utilizare: concepte, relații, reprezentare, structura documentului de descriere a cazurilor de utilizare	explicația, dialogul, exemplificarea, problematizarea	Descrierea modelului funcțional utilizând diagrame de cazuri de utilizare: concepte, relații, reprezentare, structura documentului de descriere a cazurilor de utilizare
Descrierea modelelor structurale utilizând diagrame de clase: concepte, relații, reprezentare, filtrarea informației	explicația, dialogul, exemplificarea, problematizarea	Descrierea modelelor structurale utilizând diagrame de clase: concepte, relații, reprezentare, filtrarea informației
Descrierea modelelor comportamentale utilizând diagrame de secvență și de	explicația, dialogul, exemplificarea, problematizarea	Descrierea modelelor comportamentale utilizând diagrame de secvență și de

comunicare: concepte, reprezentare, echivalența diagramelor		comunicare: concepte, reprezentare, echivalența diagramelor
Descrierea modelelor comportamentale utilizând diagrame de tranziție a stărilor. Șablonul de proiectare State	explicația, dialogul, exemplificarea, problematizarea	Descrierea modelelor comportamentale utilizând diagrame de tranziție a stărilor. Șablonul de proiectare State
Utilizarea aserțiunilor în specificarea modelelor. Design by Contract	explicația, dialogul, exemplificarea, problematizarea	Utilizarea aserțiunilor în specificarea modelelor. Design by Contract
Generarea codului pe baza modelelor UML/OCL	explicația, dialogul, exemplificarea, problematizarea	Generarea codului pe baza modelelor UML/OCL
Șabloane de testare		Șabloane de testare
8.3 Laborator	Metode de predare	Observații
Metodologii soft agile - planificarea etapelor de dezvoltare a softului. Familiarizarea cu diverse instrumente CASE UML-OCL (ex. StarUML, OCLE)	explicația, dialogul, exemplificarea, problematizarea	
Utilizarea unui instrument CASE UML pentru realizarea diagramelor de cazuri de utilizare	explicația, dialogul, exemplificarea, problematizarea	
Utilizarea unui instrument CASE UML pentru realizarea diagramei de clase aferente modelului conceptual al unui sistem	explicația, dialogul, exemplificarea, problematizarea	
Utilizarea unui instrument CASE UML pentru realizarea diagramelor de secvență/comunicare și rafinare a modelului structural	explicația, dialogul, exemplificarea, problematizarea	
Utilizarea unui instrument CASE UML pentru realizarea diagramelor de tranziție a stărilor	explicația, dialogul, exemplificarea, problematizarea	
Utilizarea unui instrument CASE UML/OCL pentru specificarea/evaluarea aserțiunilor pe modele UML	explicația, dialogul, exemplificarea, problematizarea	
Utilizarea unui instrument CASE UML/OCL pentru generarea automată a codului pe baza modelelor UML/OCL	explicația, dialogul, exemplificarea, problematizarea	
8.4 Proiect	Metode de predare	Observații
Atribuirea fiecărui student a unei aplicații de mici dimensiuni, pe care acesta va trebui să o analizeze, proiecteze, implementeze și testeze	problematizarea, exemplificarea	
Utilizarea unui instrument CASE UML și a unui editor de texte pentru realizarea modelului funcțional al aplicației. Planificarea cazurilor de utilizare pe 3 iterații. Realizarea unui prototip al interfeței grafice	problematizarea, exemplificarea	

Utilizarea unui instrument CASE UML pentru realizarea modelului conceptual al aplicației	problematizarea, exemplificarea	
Utilizarea unui instrument CASE UML și a unui IDE pentru realizarea modelului de proiectare și implementarea cazurilor de utilizare aferentei iterației 1	problematizarea, exemplificarea	
Utilizarea unui instrument CASE UML și a unui IDE pentru realizarea modelului de proiectare și implementarea cazurilor de utilizare aferentei iterațiilor 2,3	problematizarea, exemplificarea	
Testarea aplicației realizate	problematizarea, exemplificarea	
Elaborarea manualului de utilizare și predarea aplicației și a documentației aferente	problematizarea, exemplificarea	
Bibliografie		
<ul style="list-style-type: none"> • Cursul respectă recomandările curriculare IEEE / ACM pentru programele de studii de informatică. • Cursuri cu conținut similar sunt predare la majoritatea universităților din România care au programe de studii similare. <p>Companiile de dezvoltare de software consideră foarte important conținutul cursului pentru formarea viitorilor dezvoltatori de software.</p>		

9. Evaluare

Tip activitate	9.1 Criterii de evaluare ⁴	9.2 Metode de evaluare ⁵	9.3 Pondere din nota finală
9.1 Curs	Cunoașterea conceptelor și principiilor de bază ale ingineriei sistemelor soft predate	Examen scris	50%
	Insușirea conceptelor legate de modelarea softului, precum și abilitatea de a utiliza limbajul UML în acest scop		
9.2 Seminar/laborator	Cunoașterea conceptelor și principiilor de bază ale ingineriei sistemelor soft predate	Activitati seminar	10%
9.3 Laborator	Aplicarea cunoștințelor acumulate la curs/seminar/laborator pentru rezolvarea unor probleme concrete	Proiect de laborator	40%
9.6 Standard minim de promovare			
<p>Nota finală calculată cu formula dată trebuie să fie minim 5 pentru a promova examenul.</p> <ul style="list-style-type: none"> ○ Pentru promovare este OBLIGATORIE prezența la cel puțin 5 seminarii și 6 laboratoare. Studenții care nu au prezență la minimum 5 seminarii și 6 laboratoare nu se pot prezenta la examen nici în sesiunea de restanțe. Exemplu: Tema 3 cu un termen de predare în Laboratorul 4, dar predată în Laboratorul 6, primește nota maximă de 6. 			



















⁴ Criteriile de evaluare trebuie să reflecte direct rezultatele învățării vizate la nivel de program de studii, respectiv la nivel de disciplină. Mai concret, se evaluează achizițiile de învățare menționate în rezultatele anticipate ale învățării.

⁵ Se recomandă stabilirea atât a metodelor de evaluare finală, cât și a strategiei de evaluare pe parcurs.

- Nota acordată de Tutor = media aritmetică a notelor din cele 5 lucrări de laborator (acordată la finalul laboratorului 6)
- Nota acordată de coordonatorul științific = acordată în sesiune
- Nota finală = 0,5 * Nota acordată de tutore + 0,5* Nota acordată de coordonatorul științific
- Disciplina finalizata: Nota finală > = 5. Nota acordată de tutore sau nota acordată de coordonatorul științific poate fi mai mică de 5, dar nota finală trebuie să fie mai mare de 5.
- În sesiunea de restante, studentul poate preda și teme nepredate în timpul activității didactice numai dacă are cel puțin 4 prezențe. Nota acordată de tutore va fi de cel mult 6 dacă pe parcursul semestrului studentul nu a predat nicio temă. Dacă studentul a predat părți din teme în timpul semestrului, iar în sesiunea de restante a predat și alte teme, nota la fiecare temă este calculată ca și cum ar fi fost susținută în Laboratorul 6 (cu penalități corespunzătoare), dar nota finală va fi de cel mult 6.
- 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
- Participați la 75% din activitățile de seminar în timpul semestrului ȘI participați la 90% din activitățile de laborator în timpul semestrului.

Cel puțin nota 5 (de la o scară de la 1 la 10)

10. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)⁶

	Eticheta generală pentru Dezvoltare durabilă							
								
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
								Nu se aplică nici o etichetă
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Data completării:

22/05/2026

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

⁶ Selectați o singură etichetă, cea care, în conformitate cu [Procedura de aplicare a etichetelor ODD în procesul academic](#), se potrivește cel mai bine disciplinei. Dacă disciplina tratează tema dezvoltării durabile la modul general (de ex. prin prezentarea/introducerea cadrului general al dezvoltării durabile etc.) atunci se poate alocă eticheta generală de Dezvoltare Durabilă. Dacă niciuna dintre etichete nu descrie disciplina, selectați ultima opțiune: „Nu se aplică nici o etichetă”.