

FIȘA DISCIPLINEI

Inginerie Software

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	Calculatoare și tehnologia informației
1.5. Ciclu 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	E
				2.7. Regimul disciplinei	DD

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/proiect	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					
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	- Calculatoare - Instrument CASE UML - Java/.NET IDE

6.1. Competențele specifice acumulate¹

Competențe profesionale/ esențiale	- Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor - Proiectarea și integrarea sistemelor informatice utilizând tehnologii și medii de programare
Competențe transversale	- Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei - Identificarea, descrierea și derularea proceselor din managementul proiectelor, cu preluarea diferitelor roluri în echipă și descrierea clară și concisă, verbal și în scris, a rezultatelor din domeniul de activitate - Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională

6.2. Rezultatele învățării

Cunoștințe	Studentul cunoaște: - Absolventul cunoaște și înțelege conceptele, teoriile și metodele de bază ale domeniului Calculatoare și tehnologia informației și este capabil să le utilizeze în mod adecvat în comunicarea profesională. - Absolventul are cunoștințe adecvate legate de folosirea mediilor de dezvoltare integrate în scopul creării de aplicații complexe de dimensiuni mari. - Absolventul are cunoștințele necesare legate de etapele ciclului de viață al softului și a modelelor de procese software. - Absolventul cunoaște conceptele legate de modelarea softului și este capabil să implementeze cerințe funcționale și non-funcționale descrise în documente specifice pentru analiza și proiectarea sistemelor software. - Absolventul are cunoștințele necesare legate de limbajul UML, precum și abilitatea de a utiliza instrumente CASE pentru a înțelege, documenta și implementa sisteme software.
Aptitudini	Studentul este capabil să - 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. - Absolventul este capabil să dezvolte sisteme și aplicații pentru întreținerea și utilizarea de sisteme hardware, software și de comunicații. - Absolventul este capabil să realizeze testarea și evaluarea calitativă a caracteristicilor funcționale și nefuncționale ale sistemelor informatice, pe baza unor criterii specifice. - Absolventul are abilitatea de a dezvolta, proiecta și crea noi aplicații, sisteme sau produse folosind bunele practici din domeniu.
Responsabilități și autonomie	Studentul are capacitatea de a lucra independent pentru ... Absolventul are cunoștințele necesare pentru selectarea și utilizarea procedurilor potrivite de instruire pentru a facilita procesul de asimilare a cunoștințelor.

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

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.

7.1 Obiectivul general al disciplinei	• Cunoașterea și aplicarea unor concepte, principii și tehnici ingineresti solide de dezvoltare a sistemelor soft
7.2 Obiectivele specifice	• Cunoașterea etapelor ciclului de viață al softului și a modelelor de procese soft • Înțelegerea conceptelor legate de modelarea softului • Cunoașterea și aplicarea tehnicilor de dezvoltare a softului pe baza modelelor • Familiarizarea cu limbajul UML • Abilitatea de a utiliza instrumente CASE • Însușirea aspectelor de bază legate de gestiunea softului • Familiarizarea cu unele dintre metodologiile de dezvoltare, tradiționale sau agile

8. Conținuturi

8.1 Curs	Metode de predare	Observații
1. 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. Proiectarea detaliată	expunerea, conversația, discutarea unor studii de caz	
9. Proiectare sistemelor soft. Proiectarea detaliată - șabloane de proiectare	expunerea, conversația, discutarea unor studii de caz	
10. Proiectare sistemelor soft. Proiectarea detaliată - specificarea interfețelor	expunerea, conversația, discutarea unor studii de caz	
11. Implementarea sistemelor soft.	expunerea, conversația,	

Transformarea modelelor UML în cod: concepte, principii, activități, exemple	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.

8.2 Seminar	Metode de predare	Observații
1. Descrierea modelului funcțional utilizând diagrame de cazuri de utilizare: concepte, relații, reprezentare, structura documentului de descriere a cazurilor de utilizare	explicația, dialogul, exemplificarea, problematizarea	
2. Descrierea modelelor structurale utilizând diagrame de clase: concepte, relații, reprezentare, filtrarea informației	explicația, dialogul, exemplificarea, problematizarea	
3. Descrierea modelelor comportamentale utilizând diagrame de secvență și de comunicare: concepte, reprezentare, echivalența diagramelor	explicația, dialogul, exemplificarea, problematizarea	
4. Descrierea modelelor comportamentale utilizând diagrame de tranziție a stărilor. Șablonul de proiectare State	explicația, dialogul, exemplificarea, problematizarea	
5. Utilizarea aserțiunilor în specificarea modelelor. Design by Contract	explicația, dialogul, exemplificarea, problematizarea	
6. Generarea codului pe baza modelelor UML/OCL	explicația, dialogul, exemplificarea, problematizarea	
7. Șabloane de testare		
8.3 Laborator	Metode de predare	Observații
1. Metodologii soft agile - planificarea etapelor de dezvoltare a softului. Familiarizarea cu diverse	explicația, dialogul, exemplificarea, problematizarea	

instrumente CASE UML-OCL (ex. StarUML, OCLE)		
2. Utilizarea unui instrument CASE UML pentru realizarea diagramelor de cazuri de utilizare	explicația, dialogul, exemplificarea, problematizarea	
3. Utilizarea unui instrument CASE UML pentru realizarea diagramei de clase aferente modelului conceptual al unui sistem	explicația, dialogul, exemplificarea, problematizarea	
4. Utilizarea unui instrument CASE UML pentru realizarea diagramelor de secvență/comunicare și rafinare a modelului structural	explicația, dialogul, exemplificarea, problematizarea	
5. Utilizarea unui instrument CASE UML pentru realizarea diagramelor de tranziție a stărilor	explicația, dialogul, exemplificarea, problematizarea	
6. Utilizarea unui instrument CASE UML/OCL pentru specificarea/evaluarea aserțiunilor pe modele UML	explicația, dialogul, exemplificarea, problematizarea	
7. 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
1. Atribuirea fiecărui student a unei aplicații de mici dimensiuni, pe care acesta va trebui să o analizeze, proiecteze, implementeze și testeze	problematizarea, exemplificarea	
2. 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	
3. Utilizarea unui instrument CASE UML pentru realizarea modelului conceptual al aplicației	problematizarea, exemplificarea	
4. Utilizarea unui instrument CASE UML și a unui IDE pentru realizarea modelului de proiectare și implementarea cazurilor de utilizare aferentei iterației 1	problematizarea, exemplificarea	
5. 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	
6. Testarea aplicației realizate	problematizarea, exemplificarea	
7. Elaborarea manualului de utilizare și predarea aplicației și a documentației aferente	problematizarea, exemplificarea	
Bibliografie		

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

- 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.
- Companiile de dezvoltare de software consideră foarte important conținutul cursului pentru formarea viitorilor dezvoltatori de software.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs/Seminar	Cunoașterea conceptelor și principiilor de bază ale ingineriei sistemelor soft predate	Examen scris	60%
	Insușirea conceptelor legate de modelarea softului, precum și abilitatea de a utiliza limbajul UML în acest scop		
10.5 Laborator	Aplicarea cunoștințelor acumulate la curs/seminar/laborator pentru rezolvarea unor probleme concrete	Proiect de laborator	40%
10.6 Standard minim de performanță			
• minim nota 5, atât la examenul scris, cât și la evaluarea proiectului • 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.			

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

Nu se aplică.

Data completării:

Semnătura titularului de curs

Semnătura titularului de seminar

Conf. Dr. Vesca Andreea

Conf. Dr. Vesca Andreea

Data avizării în departament:

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".