

## FIȘA DISCIPLINEI

### 1. Date despre program

1.1 Instituția de învățământ superior	<b>Universitatea Babeș-Bolyai Cluj-Napoca</b>
1.2 Facultatea	<b>Facultatea de Matematică și Informatică</b>
1.3 Departamentul	<b>Departamentul de Informatică</b>
1.4 Domeniul de studii	<b>Calculatoare și Tehnologia Informației</b>
1.5 Ciclul de studii	<b>Licență</b>
1.6 Programul de studiu / Calificarea	<b>Ingineria Informației (în limba engleză)</b>

### 2. Date despre disciplină

2.1 Denumirea disciplinei (ro)		Inginerie software					
(en)		Software Engineering					
2.2 Titularul activităților de curs			Lect. dr. Vladiela Petrașcu				
2.3 Titularul activităților de seminar			Lect. dr. Vladiela Petrașcu				
2.4 Anul de studiu	3	2.5 Semestrul	6	2.6. Tipul de evaluare	E	2.7 Regimul disciplinei	Obligatorie DD
2.8 Codul disciplinei		MLE5177					

### 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	1 S 1 LP 1 P
3.4 Total ore din planul de învățământ	70	Din care: 3.5 curs	28	3.6 seminar/laborator	42
Distribuția fondului de timp:					Ore
Studiul după manual, suport de curs, bibliografie și notițe					19
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					18
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					7
Examinări					8
Alte activități: .....					
3.7 Total ore studiu individual					80
3.8 Total ore pe semestru					150
3.9 Numărul de credite					6

### 4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> <li>• Fundamentele programării</li> <li>• Programare orientată obiect</li> </ul>
4.2 de competențe	<ul style="list-style-type: none"> <li>• Abilități de programare într-un limbaj orientat-obiect de nivel înalt</li> </ul>

## 5. Condiții (acolo unde este cazul)

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

## 6. Competențele specifice acumulate

<b>Competențe profesionale</b>	<p>C4.1 Identificarea și descrierea tehnologiilor și mediilor de programare și ale conceptelor specifice ingineriei programării</p> <p>C4.2 Explicarea rolului, interacțiunii și funcționării componentelor sistemelor software</p> <p>C4.3 Elaborarea specificațiilor și proiectarea unor sisteme informatice folosind metode și instrumente specifice</p> <p>C4.4 Gestionarea ciclului de viață a sistemelor hardware, software și de comunicații pe baza evaluării performanțelor</p> <p>C4.5 Dezvoltarea și implementarea și integrarea soluțiilor software</p>
<b>Competențe transversale</b>	<p>CT1 Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei</p> <p>CT2 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</p> <p>CT3 Demonstrarea spiritului de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională</p>

## 7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"><li>• Cunoașterea și aplicarea unor concepte, principii și tehnici ingineresti solide de dezvoltare a sistemelor soft</li></ul>
7.2 Obiectivele specifice	<ul style="list-style-type: none"><li>• Cunoașterea etapelor ciclului de viață al softului și a modelelor de procese soft</li><li>• Înțelegerea conceptelor legate de modelarea softului</li><li>• Cunoașterea și aplicarea tehnicilor de dezvoltare a softului pe baza modelelor</li><li>• Familiarizarea cu limbajul UML</li><li>• Abilitatea de a utiliza instrumente CASE</li><li>• Însușirea aspectelor de bază legate de gestiunea softului</li><li>• Familiarizarea cu unele dintre metodologiile de dezvoltare, tradiționale sau agile</li></ul>

## 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. 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	
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] **Brambilla, M., Cabot, J., Wimmer, M., *Model-Driven Software Engineering in practice – 2nd edition*, Morgan and Claypool Publishers, 2017.**
- [3] **Bruegge, B., Dutoit, A., *Object-Oriented Software Engineering Using UML, Patterns and Java – 3rd ed.*, Pearson Education, 2014.**
- [4] Fowler, M. et al., *Refactoring - Improving the Design of Existing Code*, Addison Wesley, 1999.
- [5] Fowler, M., *UML Distilled: A Brief Guide to the Standard Object Modeling Language - 3rd ed.*, Addison-Wesley, 2003.
- [6] Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns*, Addison-Wesley, 1996.
- [7] Martin, R.C., *Agile Software Development: Principles, Patterns, and Practices*, Prentice Hall, 2002.
- [8] Pressman, R.S., *Software Engineering - A Practitioners Approach - 8th ed.*, McGraw-Hill, 2014.
- [9] **Seidl, M., Scholz, M., Huemer, C., Kappel, G., *UML @ Classroom: An Introduction to Object-Oriented Modeling*, Springer International Publishing, 2015.**
- [10] Schach, S.R., *Object-Oriented and Classical Software Engineering - 8th ed.*, McGraw-Hill, 2010.
- [11] Sommerville, I., *Software Engineering - 10th ed.*, Pearson, 2015.

## Links:

- [1] OMG UML 2.5.1 - [About the Unified Modeling Language Specification Version 2.5.1 \(omg.org\)](#)
- [2] OMG OCL 2.4 - [About the Object Constraint Language Specification Version 2.4 \(omg.org\)](#)
- [3] StarUML - [StarUML](#)
- [4] OCLE - [OCLE 2.0 - Object Constraint Language Environment \(ubbcluj.ro\)](#)
- [5] Eclipse Modeling Framework - [Eclipse Modeling Project | The Eclipse Foundation](#)

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	Un seminar de 2h la două săptămâni
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 <i>State</i>	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	explicația, dialogul, exemplificarea, problematizarea	

<b>8.3 Laborator</b>	<b>Metode de predare</b>	<b>Observații</b>
1. Metodologii soft agile - planificarea etapelor de dezvoltare a softului. Familiarizarea cu diverse instrumente CASE UML-OCL (ex. StarUML, OCLE)	explicația, problematizarea, exemplificarea	Un laborator de 2h la două săptămâni
2. Utilizarea unui instrument CASE UML pentru realizarea diagramelor de cazuri de utilizare	explicația, problematizarea, exemplificarea	
3. Utilizarea unui instrument CASE UML pentru realizarea diagramei de clase aferente modelului conceptual al unui sistem	explicația, problematizarea, exemplificarea	
4. Utilizarea unui instrument CASE UML pentru realizarea diagramelor de secvență/comunicare și rafinare a modelului structural	explicația, problematizarea, exemplificarea	
5. Utilizarea unui instrument CASE UML pentru realizarea diagramelor de tranziție a stărilor	explicația, problematizarea, exemplificarea	
6. Utilizarea unui instrument CASE UML/OCL pentru specificarea/evaluarea aserțiunilor pe modele UML	explicația, problematizarea, exemplificarea	
7. Utilizarea unui instrument CASE UML/OCL pentru generarea automată a codului pe baza modelelor UML/OCL	explicația, problematizarea, exemplificarea	

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	Un laborator de 2h la două săptămâni
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	
4. 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	

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

<ul style="list-style-type: none"> <li>• Cursul respectă recomandările curriculare IEEE / ACM pentru programele de studii de informatică.</li> <li>• Cursuri cu conținut similar sunt predare la majoritatea universităților din România care au programe de studii similare.</li> <li>• Companiile de dezvoltare de software consideră foarte important conținutul cursului pentru formarea viitorilor dezvoltatori de software.</li> </ul>
--

## 10. Evaluare

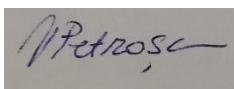
Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs/	<ul style="list-style-type: none"> <li>• Cunoașterea conceptelor și principiilor de bază ale ingineriei sistemelor soft predate</li> </ul>	Examen scris	60%

Seminar	<ul style="list-style-type: none"> <li>• Insușirea conceptelor legate de modelarea softului, precum și abilitatea de a utiliza limbajul UML în acest scop</li> </ul>		
10.5 Laborator/ Proiect	<ul style="list-style-type: none"> <li>• Aplicarea cunoștințelor acumulate la curs/seminar/laborator pentru rezolvarea unor probleme concrete</li> </ul>	Proiect de laborator	40%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> <li>• minim nota 5, atât la examenul scris, cât și la evaluarea proiectului</li> </ul>			

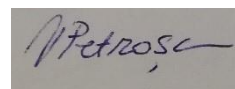
Data completării

17.05.2022

Semnătura titularului de curs



Semnătura titularului de seminar



Data avizării în departament

24.05.2022

Semnătura directorului de departament

Prof. dr. Laura Dioșan

