

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Babeș-Bolyai Cluj-Napoca
1.2 Facultatea	Matematică și Informatică
1.3 Departamentul	Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclul de studii	licență
1.6 Programul de studiu / Calificarea	Informatică linia de studiu română

2. Date despre disciplină

2.1 Denumirea disciplinei	Ingineria sistemelor soft				
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	2	2.5 Semestrul	4	2.6. Tipul de evaluare	C
				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	2
3.4 Total ore din planul de învățământ	56	Din care: 3.5 curs	28	3.6 seminar/laborator	28
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					28
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					14
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					14
Examinări					10
Alte activități:					
3.7 Total ore studiu individual	94				
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"> Fundamentele programării Programare orientată obiect
4.2 de competențe	<ul style="list-style-type: none"> Abilități de programare într-un limbaj orientat-obiect de nivel înalt

5. Condiții (acolo unde este cazul)

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

6. Competențele specifice acumulate

Competențe profesionale	<p>C2.1 Identificarea de metodologii adecvate de dezvoltare a sistemelor software</p> <p>C2.2 Identificarea și explicarea mecanismelor adecvate de specificare a sistemelor software</p> <p>C2.3 Utilizarea metodologiilor, mecanismelor de specificare și a mediilor de dezvoltare pentru realizarea aplicațiilor informatice</p> <p>C2.5 Realizarea unor proiecte informatice dedicate</p>
Competențe transversale	<p>CT1 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</p> <p>CT2 Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatice de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse</p> <p>CT3 Utilizarea unor metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, de adaptare la cerințele unei societăți dinamice și de comunicare în limba română și într-o limbă de circulație internațională</p>

7. Obiectivele disciplinei (reieșind din grila competențelor acumulate)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none"> • Cunoașterea și aplicarea unor concepte, principii și tehnici ingineresti solide de dezvoltare a sistemelor soft
7.2 Obiectivele specifice	<ul style="list-style-type: none"> • 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
<p>1. Introducere în ingineria sistemelor soft: motivație, definiție, concepte, activități</p> <p>Ciclul de viață al softului. Modele de procese soft (secvențiale, iterative, specializate)</p> <p>Tehnici de gestionare a complexității softului (abstractizare, descompunere, modelare).</p> <p>Modelarea în ingineria sistemelor soft: definiții, tipuri de modele și instrumente de modelare</p>	expunerea, conversația, discutarea unor studii de caz	
<p>2. Modelarea sistemelor soft folosind UML (Unified Modeling Language): concepte de</p>	expunerea, conversația, discutarea unor studii de caz	

modelare, tipuri de diagrame, notații, instrumente		
3. Colectarea cerințelor: concepte, activități, exemple. Modelul funcțional al unui sistem. Specificarea cerințelor nefuncționale	expunerea, conversația, discutarea unor studii de caz	
4. Analiza cerințelor: concepte, activități, exemple. Modelul conceptual al unui sistem	expunerea, conversația, discutarea unor studii de caz	
5. Proiectarea sistemelor soft: concepte, principii, activități Proiectare arhitecturala (I): Stiluri/șabloane arhitecturale	expunerea, conversația, discutarea unor studii de caz	
6. Proiectarea arhitecturala (II): Exemple	expunerea, conversația, discutarea unor studii de caz	
7. Proiectarea detaliată/obiectuală (I): principii, șabloane de proiectare	expunerea, conversația, discutarea unor studii de caz	
8. Proiectarea detaliată/obiectuală (II): șabloane de proiectare	expunerea, conversația, discutarea unor studii de caz	
9. Proiectarea detaliată/obiectuală (III): specificarea interfețelor. Metodologia <i>Design by Contract</i> Limbajul OCL (Object Constraint Language)	expunerea, conversația, discutarea unor studii de caz	
10. Implementarea sistemelor soft. Transformarea modelelor în cod: concepte, principii, activități, exemple	expunerea, conversația, discutarea unor studii de caz	
11. Verificarea și validarea sistemelor soft	expunerea, conversația, discutarea unor studii de caz	
12. Gestiunea proiectelor soft: concepte și activități	expunerea, conversația, discutarea unor studii de caz	
13. Metodologii de dezvoltare a sistemelor soft MDE (Model Driven Engineering)	expunerea, conversația, discutarea unor studii de caz	
14. Evaluare	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] Rubin, K.S., *Essential Scrum - A Practical Guide to the Most Popular Agile Process*, Addison-Wesley 2012.
- [10] Seidl, M., Scholz, M., Huemer, C., Kappel, G., *UML @ Classroom: An Introduction to Object-Oriented Modeling*, Springer International Publishing, 2015.
- [11] Schach, S.R., *Object-Oriented and Classical Software Engineering - 8th ed.*, McGraw-Hill, 2010.
- [12] 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. Colectarea cerințelor Modelul funcțional al unui sistem Diagrama UML a cazurilor de utilizare	explicația, dialogul, exemplificarea, problematizarea	Un seminar de 2 ore la două săptămâni
2. Analiza cerințelor Modelul structural de analiză (conceptual) Diagrama UML de clase	explicația, dialogul, exemplificarea, problematizarea	
3. Proiectarea obiectuală/detaliată Modele dinamice Diagrame UML de interacțiune (secvență) Rafinarea modelului structural de analiza in model structural aferent proiectarii	explicația, dialogul, exemplificarea, problematizarea	
4. Proiectarea obiectuală/detaliată Modele dinamice Diagrame UML de interacțiune (comunicare)	explicația, dialogul, exemplificarea, problematizarea	
5. Proiectarea obiectuală/detaliată Modele dinamice Diagrame UML de tranziție a stărilor Șablonul de proiectare <i>State</i>	explicația, dialogul, exemplificarea. Problematizarea	
6. Proiectarea obiectuală/detaliată <i>Design by Contract</i> : Utilizarea aserțiunilor în specificarea modelelor Limbajul OCL	explicația, dialogul, exemplificarea, problematizarea	
7. Implementare: Generarea codului pe baza modelelor UML-OCL	explicația, dialogul, exemplificarea	
8.3 Laborator	Metode de predare	Observații
1. Atribuirea, fiecarui student, a unei aplicații de mici dimensiuni, pe care acesta va trebui sa o analizeze, proiecteze, implementeze și testeze Familiarizarea cu diverse instrumente CASE UML-OCL (ex. StarUML, OCLE) si alegerea celor ce vor fi utilizate în procesul de dezvoltare	problematizarea, exemplificarea	Un laborator de 2 ore la două săptămâni
2. Utilizarea unui instrument CASE UML și a unui editor de texte pentru realizarea modelului funcțional al aplicației (diagrama cazurilor de utliiare + descrierea lor textuală, după șablon) Planificarea cazurilor de utilizare pe 3 iterații	problematizarea, exemplificarea	
3. Utilizarea unui instrument CASE UML pentru realizarea modelului conceptual Realizarea unui prototip GUI	problematizarea, exemplificarea	
4. Utilizarea unui instrument CASE UML pentru realizarea modelului de proiectare aferent iterației 1	problematizarea, exemplificarea	
5. Utilizarea unui IDE pentru implementarea/testarea cazurilor de utilizare aferentei iterației 1	problematizarea, exemplificarea	
6. Utilizarea unui instrument CASE UML pentru	problematizarea,	

realizarea modelului de proiectare aferent iterațiilor 2&3	exemplificarea	
7. Utilizarea unui IDE pentru implementarea/testarea cazurilor de utilizare aferente iterațiilor 2&3	problematizarea, exemplificarea	

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

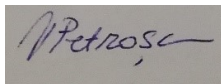
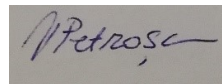
Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	<ul style="list-style-type: none"> • Cunoașterea conceptelor și principiilor de bază ale ingineriei sistemelor soft predate la curs • Insușirea conceptelor legate de modelarea softului, precum și abilitatea de a utiliza limbajul UML în acest scop 	Examen scris	60%
10.5 Seminar/laborator	<ul style="list-style-type: none"> • Aplicarea cunoștințelor acumulate la curs pentru rezolvarea unor probleme concrete 	Proiect de laborator Activitate în timpul semestrului	40%
10.6 Standard minim de performanță			
<ul style="list-style-type: none"> • minim nota 5, atât la examenul scris, cât și la evaluarea activității de laborator 			

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar

28 aprilie 2023

Data avizării în departament

Semnătura directorului de departament

.....

Prof. dr. Laura Dioșan