

## FIȘA DISCIPLINEI

### 1. Date despre program

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

### 2. Date despre disciplină

2.1 Denumirea disciplinei	<b>Ingineria sistemelor soft</b>						
2.2 Titularul activităților de curs	<b>Lect. dr. Vladiela Petrașcu</b>						
2.3 Titularul activităților de seminar	<b>Lect. dr. Vladiela Petrașcu</b>						
2.4 Anul de studiu	<b>2</b>	2.5 Semestrul	<b>4</b>	2.6. Tipul de evaluare	<b>E</b>	2.7 Regimul disciplinei	<b>obligatorie</b>

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

3.1 Număr de ore pe săptămână	<b>4</b>	Din care: 3.2 curs	<b>2</b>	3.3 seminar/laborator	<b>2</b>
3.4 Total ore din planul de învățământ	<b>56</b>	Din care: 3.5 curs	<b>28</b>	3.6 seminar/laborator	<b>28</b>
Distribuția fondului de timp:					ore
Studiul după manual, suport de curs, bibliografie și notițe					<b>14</b>
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					<b>18</b>
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					<b>30</b>
Tutoriat					<b>28</b>
Examinări					<b>4</b>
Alte activități: .....					
3.7 Total ore studiu individual	<b>94</b>				
3.8 Total ore pe semestru	<b>150</b>				
3.9 Numărul de credite	<b>6</b>				

### 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>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>C2.4 Utilizarea de criterii și metode adecvate pentru evaluarea aplicațiilor informatice</p> <p>C2.5 Realizarea unor proiecte informatice dedicate</p>
<b>Competențe transversale</b>	<p><b>CT1</b> 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><b>CT2</b> Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatiche de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse</p> <p><b>CT3</b> 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"> <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>• Însușirea aspectelor de bază legate de gestiunea softului</li> <li>• Familiarizarea cu unele dintre metodologiile de dezvoltare, tradiționale sau agile</li> <li>• Înțelegerea conceptelor legate de modelarea softului</li> <li>• Cunoașterea și aplicarea tehnicilor de dezvoltare a softului pe baza modelelor, precum și a diferitor metode de testare</li> <li>• Familiarizarea cu limbajul UML</li> <li>• Abilitatea de a utiliza instrumente CASE</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. Proiectarea sistemelor soft. Șabloane de proiectare	expunerea, conversația, discutarea unor studii de caz	
10. Proiectare sistemelor soft. Specificarea interfețelor	expunerea, conversația, discutarea unor studii de caz	
11. Implementarea sistemelor soft. Transformarea modelelor 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	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. Diagrame de cazuri de utilizare: concepte, relații, reprezentare, structura documentului de descriere a cazurilor de utilizare	explicația, dialogul, exemplificarea	Un seminar de 2 ore la două săptămâni
2. Descrierea modelelor structurale utilizând diagrame de clase: concepte, relații, reprezentare, filtrarea informației	explicația, dialogul, exemplificarea	
3. Descrierea modelelor comportamentale utilizând diagrame de secvență și de colaborare: concepte, echivalența diagramelor	explicația, dialogul, exemplificarea	
4. Descrierea modelelor comportamentale utilizând diagrame de tranziție a stărilor. Generarea codului pe baza diagramelor de tranziție a stărilor	explicația, dialogul, exemplificarea	
5. Utilizarea aserțiunilor în specificarea modelelor. Generarea codului pe baza diagramelor UML	explicația, dialogul, exemplificarea	
6. Rolul pre și post-condițiilor în specificarea interfețelor. Design by Contract	explicația, dialogul, exemplificarea	
7. Șabloane de testare	explicația, dialogul, exemplificarea	

## Bibliografie

- [1] Beck, K., *Test Driven Development*, Addison-Wesley, 2002.  
[2] Eckel, B., *Thinking in Java - 4th edition*, Prentice Hall, 2006.  
[3] Fowler, M., *UML Distilled*, Addison-Wesley, 2003.

### 8.3 Laborator

	Metode de predare	Observații
1. Metodologii soft agile - planificarea etapelor de dezvoltare a softului. 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 2 ore la două săptămâni
2. Utilizarea unui instrument CASE UML și a unui editor de texte pentru realizarea modelului funcțional al aplicației	problematizarea, exemplificarea	
3. Utilizarea unui instrument CASE UML pentru realizarea modelului conceptual	problematizarea, exemplificarea	
4. Utilizarea unui instrument CASE UML pentru realizarea modelului de proiectare	problematizarea, exemplificarea	
5. Utilizarea unui instrument CASE UML și a unui IDE pentru implementarea aplicației	problematizarea, exemplificarea	
6. Testarea aplicației realizate	problematizarea, exemplificarea	
7. Elaborarea manualului de utilizare și predarea aplicației și a documentației aferente	problematizarea, exemplificarea	

## Bibliografie

- [1] Kroll, P., Kruchten P., and Booch, G., *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*, Addison-Wesley, 2003.  
[2] Kruchten, P., *The Rational Unified Process: An Introduction - 3rd Edition*, Addison-Wesley, 2003.  
[3] Rubin, K.S., *Essential Scrum - A Practical Guide to the Most Popular Agile Process*, Addison-Wesley 2012.

## 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	Cunoașterea conceptelor și principiilor de baza ale ingineriei sistemelor soft predate la curs	Examen scris	60%
10.5 Seminar/laborator	Aplicarea cunoștințelor acumulate la curs pentru rezolvarea unor probleme concrete	Proiect de laborator Activitate în timpul seminariilor	40%
10.6 Standard minim de performanță			

- minim nota 5, atât la examenul scris, cât și la evaluarea activității de seminar/laborator

Data completării

15 aprilie 2016

Semnătura titularului de curs

Lect. dr. Vladiela PETRAȘCU

Semnătura titularului de seminar

Lect. dr. Vladiela PETRAȘCU

Data avizării în departament

.....

Semnătura directorului de departament

Prof. dr. Anca ANDREICA