

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Facultatea de Inginerie
1.3 Departamentul	Departamentul de Inginerie Electrică, Electronică și Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Licență
1.6 Programul de studii / Calificarea	Calculatoare
1.7 Forma de învățământ	IF – învățământ cu frecvență
1.8 Codul disciplinei	33

2. Date despre disciplină

2.1 Denumirea disciplinei	Proiectare software		
2.2 Titularul de curs	Șef lucrări dr ing. Marieta Gâta, Email: Marieta.GATA@ieec.utcluj.ro		
2.3 Titularul activităților de seminar / laborator / proiect	Șef lucrări dr ing. Marieta Gâta, Email: Marieta.GATA@ieec.utcluj.ro		
2.4 Anul de studiu	III	2.5 Semestrul	I
		2.6 Tipul de evaluare	E
2.7 Regimul disciplinei	Categoría formativă		DS
	Opționalitate		DI

3. Timpul total estimat

3.1 Număr de ore pe săptămână	4	din care:	3.2 Curs	2	3.3 Seminar		3.3 Laborator	1	3.3 Proiect	1
3.4 Număr de ore pe semestru	56	din care:	3.5 Curs	28	3.6 Seminar		3.6 Laborator	14	3.6 Proiect	14
3.7 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										28
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										10
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										28
(d) Tutoriat										1
(e) Examinări										2
(f) Alte activități:										-
3.8 Total ore studiu individual (suma (3.7(a)...3.7(f)))					69					
3.9 Total ore pe semestru (3.4+3.8)					125					
3.10 Numărul de credite					5					

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Programarea calculatoarelor și limbaje de programare (I și II), Programare orientată pe obiecte (I și II), Structuri de date, Proiectarea algoritmilor, Tehnici de programare fundamentale
4.2 de competente	Cunoașterea unui limbaj de programare

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Sală de curs, videoproiector, tablă, laptop Pentru cursurile online: laptop/tabletă, internet
5.2. de desfășurare a seminarului / laboratorului / proiectului	Sală de laborator cu 25 de calculatoare, Pachetele software conform programei (Microsoft Visio), internet, tablă. Pentru laboratoarele online: laptop/tabletă, internet

6. Competențele specifice acumulate

Competențe profesionale	<p>C1.1 Utilizarea adecvată în comunicarea profesională a conceptelor proprii calculabilității, complexității, paradigmelor de programare și modelării sistemelor de calcul și comunicații.</p> <p>C1.2 Utilizarea de teorii și instrumente specifice (algoritmi, scheme, modele, protocoale etc.) pentru explicarea funcționării și structurii sistemelor hardware, software și de comunicații</p> <p>C2.3 Construirea unor componente hardware, software și de comunicații folosind metode de proiectare, limbaje, algoritmi, structuri de date, protocoale și tehnologii</p> <p>C1.4 Evaluarea formală a caracteristicilor funcționale și nefuncționale ale sistemelor de calcul</p> <p>C1.5 Fundamentarea teoretică a caracteristicilor sistemelor proiectate</p> <p>C5.6 Realizarea unui proiect incluzând identificarea și analiza problemei, proiectarea, dezvoltarea și demonstrând o înțelegere a nevoii de calitate</p> <p>Modelarea unei probleme tipice ingineresti folosind aparatul formal caracteristic domeniului</p>
Competențe transversale	<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, în limba română și într-o limbă de circulație internațională, 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 specifice acumulate)

7.1 Obiectivul general al disciplinei	<p>Prezentarea principalelor tehnici și metode principale de proiectare software și deprinderea de abilitați concrete de utilizare a lor.</p> <p>În cursul semestrului studenții vor aprofunda tehnici reprezentative de proiectare care sunt utilizate la crearea proiectelor software mari. După terminarea cursului studentul va fi capabil să organizeze execuția unui proiect software, să realizeze împărțirea proiectului în componente realizând diagramele necesare pentru programatori.</p> <p>Obiective generale: dezvoltarea gândirii științifice, familiarizarea viitorului specialist cu metodele de proiectare software.</p> <p>Dobândirea abilităților necesare pentru dezvoltarea aplicațiilor folosind o abordare vizuală, orientată pe obiecte, pornind de la cunoștințele de bază de programare prezentate în cursurile predate anterior.</p>
7.2 Obiectivele specifice	<p>Proiectarea și dezvoltarea aplicațiilor de complexitate redusă și medie.</p> <p>Punctarea diferențelor esențiale, a avantajelor și a dezavantajelor folosirii diverselor tipuri de diagrame UML.</p> <p>Consolidarea abilităților de utilizare a pachetelor software Microsoft Modelling Project și www.draw.io.</p>

8. Conținuturi

8.1 Curs	Nr. ore	Metode de predare	Observații
Curs 1 Introducere Procese software Generalități	2	Expunere combinată cu prelegere cu suport pe videoproiector.	
Curs 2 Modele ale procesului de dezvoltare Definirea cerințelor utilizator Procesul de definire a cerințelor	2	Pentru cursurile online: Expunere combinată cu prelegere cu	

Curs 3 Modelarea sistemelor informatice UML Scurta prezentare UML - Sintaxa și semantica limbajului UML UML - Diagrama cazurilor de utilizare	2	suport pe laptop și tabletă în Meeting din Microsoft Teams.		
Curs 4 UML - Diagrame de activitate	2			
Curs 5 Paradigma programării orientate obiect Concepte și principii ale programării orientate obiect Analiză și proiectare orientată obiect UML - Diagrame de clase și diagrame de obiecte	2			
Curs 6 UML - Diagrame de clase și diagrame de obiecte	2			
Curs 7 UML - Diagrame de secvențe	2			
Curs 8 UML - Diagrame de colaborare	2			
Curs 9 UML - Diagrame de stări	2			
Curs 10 UML - Diagrame de componente UML - Diagrame de desfășurare UML - Studii de caz	2			
Curs 11 Limbajul XML Sintaxa documentelor XML: elemente, atribute, comentarii	2			
Curs 12 Validarea documentelor XML 1. Validarea documentelor XML cu DTD 2. Validarea documentelor XML cu XSD Schema	2			
Curs 13 Document Object Model (DOM)	2			
Curs 14 Procesarea documentelor XML Formatarea documentelor XML utilizând XSLT	2			
Bibliografie 1. <i>Beginning XML</i> , D. Hunter, J. Rafter, J. Fawcett, E. Vlist , 4th Edition (Programmer to Programmer), 2007 2. <i>XSLT Cookbook: Solutions and Examples for XML and XSLT Developers</i> , Sal Mangano, 2nd Edition, O'Reilly 3. <i>Practical Approach To Software Engineering: With UML Diagram</i> , Sunny Singh, Satvik Vats, Anurag Vyas, LAP LAMBERT Academic Publishing, 2015 4. <i>Use Case Driven Object Modeling with UML: Theory and Practice</i> , Doug Rosenberg, Matt Stephens, Apress, 2nd Edition, 2013 5. <i>Using XML</i> , Lee Anne Philips, Editura Teora 2001 6. <i>The Diagrams of UML 2.0</i> , Scott Ambler, http://www.agilemodeling.com/essays/umlDiagrams.htm , 2003 7. <i>Learning UML 2.0</i> , Hamilton, K., Miles, R. , O'Reilly, 2006 8. <i>UML Distilled: A Brief Guide to the Standard Object Modeling Language</i> , Fowler, M., 3rd Edition, Addison-Wesley Professional, 2003 9. <i>Design Patterns: Elements of Reusable Object-Oriented Software</i> , Kevin Zhang, http://www.uml .				

org.cn/c++/pdf/DesignPatterns.pdf			
8.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
Laborator 1 Introducere Generalități Prezentare Microsoft Visio	2	Modelare, problematizare, studiu de caz. Realizarea pe calculator a tuturor tipurilor de diagrame UML utilizând Microsoft Visio. Pentru laboratoarele online: folosirea facilităților oferite de Meeting în Microsoft Teams, Office 365, Whiteboard, OneNote, Forms, SharePoint.	
Laborator 2 Definirea cerințelor utilizator Crearea unei diagrame de clasă în NETBeans	2		
Laborator 3 UML - Diagrama cazurilor de utilizare	2		
Laborator 4 UML - Diagrame de activitate	2		
Laborator 5 UML - Diagrame de clase și diagrame de obiecte	2		
Laborator 6 UML - Diagrame de clase și diagrame de obiecte	2		
Laborator 7 UML - Diagrame de secvențe	2		
Laborator 8 UML - Diagrame de colaborare	2		
Laborator 9 UML - Diagrame de stări	2		
Laborator 10 UML - Diagrame de componente și diagrame de desfășurare	2		
Laborator 11 Limbajul XML Sintaxa documentelor XML: elemente, atribute, comentarii	2		
Laborator 12 Validarea documentelor XML 1. Validarea documentelor XML cu DTD 2. Validarea documentelor XML cu XSD Schema	2		
Laborator 13 Document Object Model (DOM)	2		
Laborator 14 Procesarea documentelor XML Formatarea documentelor XML utilizând XSLT	2		
Bibliografie 1. <i>Beginning XML</i> , D. Hunter, J. Rafter, J. Fawcett, E. Vlist , 4th Edition (Programmer to Programmer), 2007 2. <i>XSLT Cookbook: Solutions and Examples for XML and XSLT Developers</i> , Sal Mangano, 2nd Edition, O'Reilly 3. <i>Practical Approach To Software Engineering: With UML Diagram</i> , Sunny Singh, Satvik Vats, Anurag Vyas, LAP LAMBERT Academic Publishing, 2015 4. <i>Use Case Driven Object Modeling with UML: Theory and Practice</i> , Doug Rosenberg, Matt Stephens, Apress, 2nd Edition, 2013 5. <i>Using XML</i> , Lee Anne Philips, Editura Teora 2001 6. <i>The Diagrams of UML 2.0</i> , Scott Ambler, http://www.agilemodeling.com/essays/umlDiagrams.h tm , 2003			

7. *Learning UML 2.0*, Hamilton, K., Miles, R. , O'Reilly, 2006
 8. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Fowler, M., 3rd Edition, Addison-Wesley Professional, 2003
 9. Design Patterns: Elements of Reusable Object-Oriented Software, Kevin Zhang, <http://www.uml.org.cn/c++/pdf/DesignPatterns.pdf>

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

Cursul oferă ajutor pentru cursurile din profilul software al specializării Calculatoare.
 Cursul creează legături directe/imediate cu următoarele discipline deja predate: Programarea calculatoarelor și limbaje de programare (I și II), Programare orientată pe obiecte (I și II), Structuri de date, Proiectarea algoritmilor, Tehnici de programare fundamentale.
 Utilitatea pentru piața muncii: aplicațiile de laborator sunt realizate cu ajutorul tehnologiei Microsoft Visio și oferă cunoștințe în această direcție.
 Cunoștințele de programare care folosesc conceptele UML și XML sunt importante pentru cei care doresc să se angajeze ca programatori, deoarece mulți dintre angajatorii reprezentativi din domeniul aferent programului de studiu solicită cunoașterea tehnologiilor și conceptelor din UML și XML.

10. Evaluare (prezența fizică / online)

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoștințe teoretice însușite (cantitatea, corectitudinea, acuratețea)	Evaluarea finală Rezolvarea unui test grilă (evaluare scrisă), realizarea unor diagrame UML și descrierea unui scenariu pentru o problemă dată	50% (minim 5)
10.5 Seminar/Laborator	Frecvența/relevanța intervențiilor. Cunoașterea modului de utilizare a instrumentelor specifice. Evaluarea unor instrumente sau realizări, prelucrarea și interpretarea unor rezultate.	Portofoliu de lucrări/aplicații care sunt verificate la finalul fiecărei ore de laborator Proiect -Prezentarea proiectului (care constă din realizarea diagramelor UML pentru o aplicație dată) -Autoevaluarea critică a proiectului	10% (minim 5) 40% (minim 5)

10.6 Standard minim de performanță

În orele de laborator sunt propuse un număr de trei aplicații în fiecare laborator pe care studentul le va rezolva (în total, la finalul orelor de laborator, studentul va trebui să aibă rezolvate un număr de minim 30 aplicații pentru nota 10, pentru nota 5 este necesar să fie rezolvate 15 aplicații). Programele propuse la laborator trebuie să rezolve toate cerințele impuse în aplicații iar diagramele nu trebuie să aibă erori.
 În orele de proiect studentul va lucra la proiectul pe care și l-a ales (vor exista un număr de teme propuse de către profesor, fiecare student va avea o altă temă aleasă din cele propuse de către profesor sau propusă de către student). Proiectul pe care studentul va trebui să-l realizeze va impune proiectarea unei aplicații de complexitate medie. Proiectul va conține toate diagramele UML prezentate la curs. Pentru nota 10 proiectul va trebui să rezolve toate cerințele impuse și să fie de dificultate înaltă. Pentru nota 5 proiectul va trebui să rezolve cel puțin 50% din cerințele impuse și să fie de dificultate joasă.
 Dobândirea unei gândiri analitice și a unor abilități de generalizare independente de limbajul de programare folosit, sunt o componentă fundamentală proiectării oricărei aplicații software complexe.
 Dobândirea unor abilități practice legate de particularizarea diagramelor UML pentru o aplicație dată.

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
	Curs	Şef lucrări dr. ing. Marieta GÂTA	
	Aplicații	Şef lucrări dr. ing. Marieta GÂTA	

Data avizării în Consiliul Departamentului _____	Director Departament, Şef lucrări dr. ing. Claudiu Ionel Lung
Data aprobării în Consiliul Facultății, _____	Decan, Conf. de. ing. Dinu DARABA