

FIȘA DISCIPLINEI¹

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnică Timișoara
1.2 Facultatea ² / Departamentul ³	Facultatea de Inginerie Hunedoara / Inginerie Electrică și Informatică Industrială
1.3 Catedra	—
1.4 Domeniul de studii (denumire/cod ⁴)	ȘTIINȚE INGINEREȘTI APLICATE / 270
1.5 Ciclul de studii	Licenta
1.6 Programul de studii (denumire/cod/calificarea)	INFORMATICĂ INDUSTRIALĂ / 50 / Inginer

2. Date despre disciplină

2.1 Denumirea disciplinei/Categoria formativă ⁵	Inginerie software / DD						
2.2 Titularul activităților de curs	Ș. L. Dr. Iordan Anca-Elena						
2.3 Titularul activităților aplicative ⁶	Ș. L. Dr. Ing. Abrudean Cristian						
2.4 Anul de studii ⁷	IV	2.5 Semestrul	II	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei ⁸	DO

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)⁹

3.1 Număr de ore asistate integral/săptămână	3,5 , format din:	3.2 ore curs	2	3.3 ore seminar/laborator/proiect	1,5
3.1* Număr total de ore asistate integral/sem.	49 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	21
3.4 Număr de ore asistate parțial/săptămână	, format din:	3.5 ore practică		3.6 ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/semestru	, format din:	3.5* ore practică		3.6* ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/săptămână	3,92 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1,92
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1
3.7* Număr total de ore activități neasistate/semestru	55 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			14
		ore studiu individual după manual, suport de curs, bibliografie și notițe			27
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			14
3.8 Total ore/săptămână ¹⁰	7,42				
3.8* Total ore/semestru	104				
3.9 Număr de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	<ul style="list-style-type: none"> Programarea calculatoarelor 1, Programarea calculatoarelor 2, Proiectarea algoritmilor, Programarea calculatoarelor și limbaje de programare, Baze de date
4.2 de competențe	<ul style="list-style-type: none">

¹ Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 și cerințelor Standardelor specifice ARACIS valabile începând cu 01.10.2017.

² Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina.

³ Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului.

⁴ Se înscrie codul prevăzut în HG nr.140/16.03.2017 sau în HG similare actualizate anual.

⁵ Disciplina se încadrează potrivit planului de învățământ în una dintre următoarele categorii formative: disciplină fundamentală (DF), disciplină de domeniu (DD), disciplină de specialitate (DS) sau disciplina complementară (DC).

⁶ Prin activități aplicative se înțeleg activitățile de: seminar (S) / laborator (L) / proiect (P) / practică (Pr).

⁷ Anul de studii în care este prevăzută disciplina în planul de învățământ.

⁸ Disciplina poate avea unul din următoarele regimuri: disciplină impusă (DI), disciplină opțională (DO) sau disciplină facultativă (Df).

⁹ Numărul de ore de la rubricile 3.1*, 3.2*,...,3.8* se obțin prin înmulțirea cu 14 (săptămâni) a numărului de ore din rubricile 3.1, 3.2,...., 3.8. Informațiile din rubricile 3.1, 3.4 și 3.7 sunt chei de verificare folosite de ARACIS sub forma: (3.1)+(3.4) ≥ 28 ore/săpt. și (3.8) ≤ 40 ore/săpt.

¹⁰ Numărul total de ore / săptămână se obține prin însumarea numărului de ore de la punctele 3.1, 3.4 și 3.7.

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	<ul style="list-style-type: none">• Sală de curs echipată cu videoproiector și conexiune la Internet.• Studenții nu se vor prezenta la prelegeri cu telefoanele mobile deschise.
5.2 de desfășurare a activităților practice	<ul style="list-style-type: none">• Sală de laborator echipată cu videoproiector și computere.• Studenții nu se vor prezenta la activitățile practice cu telefoanele mobile deschise.

6. Competențe la formarea cărora contribuie disciplina

Competențe specifice	<p>C 2.</p> <p>C 2.1. Descrierea structurii și a modului de funcționare a sistemelor informatice în general; C 2.2. Explicarea rolului, funcționalității și utilității sistemelor informatice în general și a sistemelor de prelucrare și gestiune a datelor în domeniul specializării; C 2.3. Utilizarea componentelor software ale sistemelor informatice, folosind algoritmi, protocoale, limbaje, structuri de date; C 2.4. Aprecierea caracteristicilor și calității sistemelor informatice; C 2.5. Prelucrarea și gestionarea datelor utilizând sisteme informatice dedicate.</p> <p>C 6.</p> <p>C 6.1. Descrierea principiilor de bază privind achiziția și transmisia de date din proces; C 6.2. Explicarea rolului componentelor sistemelor de achiziție de date aferente unui sistem informatic destinat conducerii automate a proceselor industriale; C 6.3. Configurarea sistemelor de achiziție și transmisie de date aferente proceselor industriale; C 6.4. Utilizarea adecvată a metodelor de evaluare a performanțelor sistemelor informatice și de validare a datelor achiziționate din proces; C 6.5. Implementarea componentelor sistemelor informatice de achiziție de date.</p> <ul style="list-style-type: none">•
Competențele profesionale în care se înscriu competențele specifice	<ul style="list-style-type: none">• C 2. Utilizarea sistemelor informatice de prelucrare și gestiune a datelor.• C 6. Configurarea, implementarea și folosirea sistemelor de achiziție de date.
Competențele transversale în care se înscriu competențele specifice	<ul style="list-style-type: none">•

7. Obiectivele disciplinei (asociate competențelor de la punctul 6)

7.1 Obiectivul general al disciplinei	<ul style="list-style-type: none">• Scopul acestei discipline îl reprezintă însușirea principalelor concepte, metode și tehnici ale ingineriei software, cu accent pe reutilizarea codului sursă atât din punct de vedere al programării orientate pe obiect, cât și din punct de vedere al programării generice în limbajele de programare C++ și C#.
7.2 Obiectivele specifice	<p>Obiectivele specifice ale acestei discipline sunt:</p> <ul style="list-style-type: none">• Cunoașterea etapelor ciclului de viață al unui sistem soft;• Înțelegerea conceptelor legate de modelarea softului;• Familiarizarea cu limbajul de modelare UML;• Abilitatea de a utiliza instrumente CASE.

8. Conținuturi¹¹

8.1 Curs	Număr de ore	Metode de predare ¹²
1. Fazele dezvoltării unui sistem soft 1.1. Ciclul de viață al softului. Ciclul de viață V 1.2. Faza de analiză 1.3. Faza de proiectare 1.4. Faza de implementare 1.5. Faza de testare	2	Studenții au acces la curs în format electronic. Se vor utiliza atât prezentări interactive cât și tradiționale. Se vor folosi: problematizarea, studiu de caz, conversația.
2. Limbajul de modelare unificat UML 2.1. Caracteristicile și avantajele UML 2.2. Tipuri de diagrame 2.3. Instrumente CASE: IBM Rational Rhapsody, Microsoft Visio, ArgoUML, StarUML 2.4. Diagrame specifice fazei de analiză 2.5. Diagrame specifice fazei de proiectare 2.6. Diagrame specifice fazei de implementare	4	
3. Analiza sistemelor soft 3.1. Colectarea și analiza cerințelor 3.2. Instrumente utilizate în specificarea cerințelor. IBM Rational DOORS 3.3. Metode de analiză orientate pe obiecte	4	
4. Proiectarea sistemelor soft în C++ și C# 4.1. Caracteristicile unui proiectării orientate pe obiecte 4.2. Metode de proiectare 4.3. Principii de proiectare 4.3.1. Principiul deschis-închis 4.3.2. Principiul substituției 4.3.3. Principiul inversiunii dependențelor 4.3.4. Principiul responsabilității unice 4.3.5. Principiul segregării interfețelor 4.4. Șabloane arhitecturale 4.5. Șabloane de proiectare 4.5.1. Șabloane creaționale 4.5.2. Șabloane structurale 4.5.3. Șabloane comportamentale	10	
5. Verificarea și validarea sistemelor soft 5.1. Testarea programelor 5.1.1. Criterii de alegere a datelor de test 5.1.2. Testarea white-box 5.1.3. Testarea black-box 5.2. Instrumente de testare: QA-C++, csUnit, NUnit, xUnit.net 5.3. Verificarea modelelor, Verificarea specificației și a corectitudinii algoritmilor 5.4. Testarea modulelor și a sistemului final 5.5. Verificarea documentației 5.6. Validarea programelor 5.7. Certificarea produselor soft	6	
6. Managementul unui proiect software 6.1. Managementul software 6.2. Managementul configurației 6.3. Managementul echipei 6.4. Instrumente integrate de management al proiectelor soft: JIRA, MKS Integrity 6.5. Estimarea costului unui sistem soft	2	

¹¹ Se detaliază toate activitățile didactice prevăzute prin planul de învățământ (tematicile prelegerilor și ale seminariilor, lista lucrărilor de laborator, conținuturile etapelor de elaborare a proiectelor, tematica fiecărui stagiu de practică). Titlurile lucrărilor de laborator care se efectuează pe standuri vor fi însoțite de notația „(*)”.

¹² Prezentarea metodelor de predare va include și folosirea noilor tehnologii (e-mail, pagină personalizată de web, resurse în format electronic etc.).

Bibliografie¹³

1. Dorin Bocu, Răzvan Bocu, Provocări și metode de abordare în managementul proiectelor IT, Editura Albastră, Cluj-Napoca, 2013
2. Dorin Bocu, Răzvan Bocu, Modelare obiect orientată cu UML, Editura Albastră, Cluj-Napoca, 2006
3. Mircea Cezar Preda, Ana-Maria Mirea, Doina Lavinia Preda, Constantin Teodorescu-Mihai, Introducere în programarea orientată-obiect. Concepte fundamentale din perspectiva ingineriei software, Editura Polirom, Iași, 2010
4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Șabloane de proiectare. Elemente de software reutilizabil orientat pe obiect, Editura Teora, 2002
5. Alan Ezust, Paul Ezust, Introduction to Design Patterns in C++ with Qt, Prentice Hall, 2011
6. Steven John Metsker, Design Patterns in C#, Addison-Wesley Professional, 2004
7. Pankaj Jalote, A Concise Introduction to Software Engineering, Springer-Verlag Londra, 2008

8.2 Activități aplicative ¹⁴	Număr de ore	Metode de predare
1. Elaborarea diagramelor specifice fazei de analiză: diagrama cazurilor de utilizare și diagrame de activități utilizând ArgoUML IBM Rational Rhapsody și Visual Studio	3	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
2. Elaborarea diagramelor specifice fazei de proiectare: diagrama de clase, diagrama de pachete, diagrame de secvență, diagrame de colaborare și diagrame de stare utilizând ArgoUML, IBM Rational Rhapsody și Visual Studio	3	
3. Elaborarea diagramelor specifice fazei de implementare: diagrama de componente utilizând ArgoUML, IBM Rational Rhapsody și Visual Studio	3	
4. Analiza, proiectarea și implementarea de aplicații în C++ și C# ce utilizează șabloane creaționale	3	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
5. Analiza, proiectarea și implementarea de aplicații în C++ și C# ce utilizează șabloane structurale	3	
6. Analiza, proiectarea și implementarea de aplicații în C++ și C# ce utilizează șabloane comportamentale	3	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
7. Testare folosind instrumentele: QA-C++, csUnit, NUnit, xUnit.net	3	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.

Bibliografie¹⁵

1. Anca Daniela Ioniță, Modelarea UML în ingineria sistemelor de programare, Editura All, București, 2003
2. Tony Bevi, C# Design Pattern Essentials, Ability First Limited, 2012
3. Roy Osherove, The Art of Unit Testing with examples in C#, Manning Publications, 2013
4. Bender James, McWherter Jeff, Professional Test Driven Development with C#, Wrox, 2014
5. Jeff Langr, Modern C++ Programming with Test-Driven Development, 2013
6. Samek Miro, Practical UML Statecharts in C/C++, Newnes, 2013

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

- Disciplina vine în întâmpinarea așteptărilor angajatorilor reprezentativi din domeniul aferent programului prin conținutul orelor de curs și laborator.

10. Evaluare

¹³ Cel puțin un un titlu trebuie să aparțină colectivului disciplinei iar cel puțin un titlu trebuie să se refere la o lucrare de referință pentru disciplină, de circulație națională și internațională, existentă în biblioteca UPT.

¹⁴ Tipurile de activități aplicative sunt cele precizate în nota de subsol 5. Dacă disciplina conține mai multe tipuri de activități aplicative atunci ele se trec consecutiv în liniile tabelului de mai jos. Tipul activității se va înscrie într-o linie distinctă sub forma: „Seminar:”, „Laborator:”, „Proiect:” și/sau „Practică:”.

¹⁵ Cel puțin un titlu trebuie să aparțină colectivului disciplinei.

Tip activitate	10.1 Criterii de evaluare ¹⁶	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoștințe teoretice	Oral - subiecte teoretice și aplicații	0,66
10.5 Activități aplicative	S:		
	L: Abilități în analiza, proiectarea și implementarea aplicațiilor de laborator	Oral – aplicații utilizând calculatorul	0,34
	P¹⁷:		
	Pr:		
10.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁸)			
<ul style="list-style-type: none"> La finalul cursului, respectiv al laboratorului, studenții trebuie să aibă cunoștințe teoretice și abilități de cercetare, strict necesare viitorilor specialiști dovedind competențe în analiza, proiectarea, implementarea și testarea sistemelor informatice dezvoltate în limbajele de programare C++ și C#. 			

Data completării

04.09.2017

**Director de departament
(semnătura)**

.....

**Titular de curs
(semnătura)**

.....

Data avizării în Consiliul Facultății¹⁹

06.09.2017

**Titular activități aplicative
(semnătura)**

.....

**Decan
(semnătura)**

.....

¹⁶ Fișele disciplinelor trebuie să conțină procedura de evaluare a disciplinei cu precizarea criteriilor, a metodelor și a formelor de evaluare, precum și cu precizarea ponderilor atribuite acestora în nota finală. Criteriile de evaluare se formulează în mod distinct pentru fiecare activitate prevăzută în planul de învățământ (curs, seminar, laborator, proiect). Ele se vor referi și la formele de verificare pe parcurs (teme de casă, referate ș.a.)

¹⁷ În cazul când proiectul nu este o disciplină distinctă, în această rubrică se va preciza și modul în care rezultatul evaluării proiectului condiționează admiterea studentului la evaluarea finală din cadrul disciplinei.

¹⁸ Nu se va explica cum se acorda nota de promovare.

¹⁹ Avizarea este precedată de discutarea punctului de vedere al board-ului de care aparține programul de studii cu privire la fișa disciplinei.