FIŞA DISCIPLINEI[1]

1. Date despre program

1.1 Instituţia de învăţământ 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])	ŞTIINŢE INGINEREŞTI 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]			Ingineria sistemelor de programe / DD				
2.2 Titularul activităţilor de curs			Șef lucr. Dr. Ing. Ghiormez Loredana				
2.3 Titularul activităţilor aplicative[6]			Șef lucr. Dr. Ing. Ghiormez Loredana				
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	3,64 , 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			1,64
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1
3.7* Număr total de ore activități neasistate/ semestru	51 , 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			23
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			14
3.8 Total ore/săptămână [10]	7,14				
3.8* Total ore/semestru	100				
3.9 Număr de credite	4				

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	• Programarea calculatoarelor 1, Programarea calculatoarelor 2, Proiectarea algoritmilor, Programarea calculatoarelor și limbaje de programare, Baze de date,
4.2 de competenţe	•

[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), discipină 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. Condiţii (acolo unde este cazul)

5.1 de desfăşurare a cursului	• Sală de curs echipată cu videoproiector și conexiune la Internet. • Studenții nu se vor prezenta la prelegeri cu telefoanele mobile deschise.
5.2 de desfăşurare a activităţilor practice	• Sală de laborator echipată cu cu videoproiector și computere. • Studenții nu se vor prezenta la activitățile practice cu telefoanele mobile deschise.

6. Competenţe la formarea cărora contribuie disciplina

Competenţe specifice	C 2. C 2.1. Descrierea structurii şi a modului de funcţionare a sistemelor informatice în general; C 2.2. Explicarea rolului, funcţionalităţii şi utilităţii sistemelor informatice în general şi a sistemelor de prelucrare şi gestiune a datelor în domeniul specializării; C 2.3. Utilizarea componentelor software ale sistemelor informatice, folosind algoritimi, protocoale, limbaje, structuri de date; C 2.4. Aprecierea caracteristicilor şi calităţii sistemelor informatice; C 2.5. Prelucrarea şi gestionarea datelor utilizând sisteme informatice dedicate. C 6. C 6.1. Descrierea principiilor de bază privind achiziţia şi transmisia de date din proces; C 6.2. Explicarea rolului componentelor sistemelor de achiziţie de date aferente unui sistem informatic destinat conducerii automate a proceselor industriale; C 6.3. Configurarea sistemelor de achiziţie şi transmisie de date aferente proceselor industriale; C 6.4. Utilizarea adecvată a metodelor de evaluare a performanţelor sistemelor informatice şi de validare a datelor achiziţionate din proces; C 6.5. Implementarea componentelor sistemelor informatice de achiziţie de date. •
Competențele profesionale în care se înscriu competențele specifice	• C 2. Utilizarea sistemelor informatice de prelucrare şi gestiune a datelor. • C 6. Configurarea, implementarea şi folosirea sistemelor de achiziţie de date.
Competențele transversale în care se înscriu competențele specifice	•

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

7.1 Obiectivul general al disciplinei	• Scopul acestei discipline îl reprezintă însuşirea principalelor concepte, metode şi tehnici ale ingineriei sistemelor de programare, cu accent pe reutilizarea codului sursă atât din punct de vedere al programării orientate pe obiect, cât şi din punct de vedere al programării generice în limbajele de programare C++ și C#.
7.2 Obiectivele specifice	Obiectivele specifice ale acestei discipline sunt: • Însușirea cunoştinţelor despre metodologiile de realizare a sistemelor soft; • Familiarizarea cu conceptele şi preocupările moderne în scrierea softului de aplicaţie; • Însușirea cunoştinţelor despre tipurile de modele şi instrumentele de modelare folosite în dezvoltarea sistemelor de programare.

8. Conţinuturi[11]

8.1 Curs	Număr de ore	Metode de predare[12]
1. Etapele realizării unui sistem de programare 1.1. Ce se înțelege prin ingineria sistemelor de programare? 1.2. Ciclul de viaţă al sistemului de programare, tipuri de abordări în rezolvarea de probleme 1.3. Etapa de analiză 1.4. Etapa de proiectare 1.5. Etapa de implementare 1.6. Etapa 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. Metodologii de dezvoltare a programelor 2.1. Metodologii generice 2.2. Metodologii concrete	2	
3. Managementul unui sistem de programare 3.1. Funcţiile managementului 3.2. Managementul software 3.3. Managementul configuraţiei 3.4. Managementul echipei 3.5. Instrumente integrate de management al proiectelor soft: JIRA, MKS Integrity	4	
4. Estimarea costului unui sistem de programare 4.1. Modele algoritmice clasice 4.2. Modele algoritmice moderne 4.3. Distribuirea forţei de muncă în timp	2	
5. Analiza şi proiectarea orientată pe obiecte a sistemelor de programare 5.1. Principiile analizei orientate pe obiecte. 5.2. Metode de analiză orientată obiect. Analiza cerințelor utilizând UML 5.3. Metode de analiză orientate pe obiect 5.4. Caracteristicile unei proiectări orientate pe obiecte corecte 5.5. Etapele proiectării orientate pe obiecte. Diagrame UML specifice fazei de proiectare	6	
6. Reutilizarea resurselor de programare 6.1. Reutilizarea produselor intermediare 6.2. Reutilizarea instrumentelor 6.3. Programarea generică în C++ și C#. Colecții generice de date 6.4. Șabloane de proiectare în C++ și C#	8	
7. Verificarea şi validarea sistemelor de programare 7.1. Testarea programelor. Criterii de alegere a datelor de test 7.2. Instrumente de testare: QA-C++, csUnit, NUnit, xUnit.net 7.3. Verificarea modelelor. Inspectarea tuturor subproduselor 7.4. Testarea modulelor şi a sistemului de programare final 7.5. Verificarea documentaţiei 7.6. Validarea programelor 7.7. Certificarea produselor de programare. Planificarea verificării şi validării	4	

[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.).

Bibliografie[13]

1. Ghiormez L., Curs în format electronic – Ingineria sistemelor de programe, Campusul virtual al UPT, https://cv.upt.ro/course/view.php?id=2716
2. Sabin Goron, Elemente de ingineria produselor software, Editura Risoprint, Cluj-Napoca, 2006
3. Dorin Bocu, Iniţiere în ingineria sistemelor soft, Editura Albastră, Cluj-Napoca, 2002
4. Militon Frenţiu, Verificarea și validarea sistemelor soft, Presa Universitară Clujană, Cluj-Napoca, 2010
5. Philip Wadler, Maurice Naftali, Java Generics and Collections, O'Reilly Media, 2013
6. John Hunt, Guide to the Unified Process Featuring UML, Java and Design Patterns, Springer London Ltd, 2014
7. Hamill Paul, Unit Test Frameworks, O'Reilly Media, 2016

8.2 Activităţi aplicative[14]	Număr de ore	Metode de predare
1. Elaborarea diagramelor specifice fazei de analiză: diagrama cazurilor de utilizare utilizând ArgoUML IBM Rational Rhapsody şi Visual Studio	4	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
2. Elaborarea diagramelor specifice fazei de analiză: diagrame de activităţi utilizând ArgoUML IBM Rational Rhapsody şi Visual Studio	4	
3. Elaborarea diagramelor specifice fazei de proiectare: diagrama de clase, diagrama de pachete utilizând ArgoUML IBM Rational Rhapsody şi Visual Studio	4	
4. Elaborarea diagramelor specifice fazei de proiectare: diagrame de stare utilizând ArgoUML IBM Rational Rhapsody şi Visual Studio	2	
5. Elaborarea diagramelor specifice fazei de proiectare: diagrame de secvenţă, diagrame de colaborare utilizând ArgoUML IBM Rational Rhapsody şi Visual Studio	2	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
6. Elaborarea diagramelor specifice fazei de implementare: diagrama de componente utilizând ArgoUML IBM Rational Rhapsody şi Visual Studio	2	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
7. Analiza, proiectarea şi implementarea de aplicaţii în limbajul C++ şi C# ce utilizează şablonul arhitectural MVC. Testarea aplicațiilor folosind mediul de programare Visual Studio, respectiv instrumentele QA-C++, csUnit, NUnit, xUnit.net. Recuperări de laborator.	3	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.

Bibliografie[15]

1. Ghiormez L., Laborator în format electronic – Ingineria sistemelor de programe, Campusul virtual al UPT, https://cv.upt.ro/course/view.php?id=2716
2. Anca Daniela Ioniță, Modelarea UML în ingineria sistemelor de programare, Editura All, București, 2003
3. Tony Bevi, C# Design Pattern Essentials, Ability First Limited, 2012
4. Roy Osherove, The Art of Unit Testing with examples in C#, Manning Publications, 2013
5. Bender James, McWherter Jeff, Professional Test Driven Development with C#, Wrox, 2014
6. Jeff Langr, Modern C++ Programming with Test-Driven Development, 2013
7. Samek Miro, Practical UML Statecharts in C/C++, Newnes, 2013

9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi 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

Tip activitate	**10.1** Criterii de evaluare[16]	**10.2** Metode de evaluare	**10.3** Pondere din nota finală
10.4 Curs	Cunoștințe teoretice	Scris - subiecte teoretice și aplicații.	0,66

[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.
[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.)

		Rezolvările se încarcă pe campusul virtual UPT sau se scriu pe foaie. Examenul se desfășoară prin intermediul Campusului Virtual UPT sau se primesc subiecte tipărite. În caz de scenariu online se utilizează și aplicația de videoconferință (Zoom)	
10.5 Activităţi aplicative	**S:**		
	L: Abilități în analiza, proiectarea și implementarea aplicațiilor de laborator	Oral – aplicații utilizând calculatorul In caz de scenariu online testele de control se desfășoară prin intermediul campusului virtual UPT	0,34
	P[17]**:**		
	Pr:		

10.6 Standard minim de performanţă (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor[18])

- Nota 5 la activitățile aplicative constă în implementarea a cel puțin 4 diagrame UML implementate pe aceeași tematică, dintre care una obligatoriu să fie diagrama de clase.
- Nota 5 la examenul final se obține în condițiile obținerii a minim jumătate din punctajul total alocat subiectelor.
- Pentru a promova disciplina trebuie ca ambele activități (curs și laborator) să fie promovate.

Data completării

04.10.2022

Titular de curs
(semnătura)

……

Titular activităţi aplicative
(semnătura)

……

Director de departament
(semnătura)

……

Data avizării în Consiliul Facultăţii[19]

18.10.2022

Decan
(semnătura)

……

[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.