FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea [1] / Departamentul[2]	Robotică, Inteligență Artificială și Sisteme Autonome / Calculatoare și Tehnologia Informației
1.3 Domeniul de studii (denumire/cod[3])	Informatică / 10
1.4 Ciclul de studii	Master
1.5 Programul de studii (denumire/cod/calificarea)	Tehnologii Informatice / 10 / Tehnologii Informatice

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă[4]			Mentenanta aplicatiilor industriale / DF				
2.1b Denumirea disciplinei în limba engleză			Industrial Application Maintenance				
2.2 Titularul activităţilor de curs			Cristina MARINESCU				
2.3 Titularul activităţilor aplicative[5]			Cristina MARINESCU				
2.4 Anul de studiu[6]	1	**2.5** Semestrul	1	**2.6** Tipul de evaluare	E	**2.7** Regimul disciplinei[7]	DO

3. Timp total estimat - ore pe semestru (activități directe (asistate integral), activități asistate parțial și activități neasistate[8])

3.1 Număr de ore asistate integral/săptămână	3.5, din care:	ore curs	1.5	ore seminar/laborator/proiect			2
3.1* Număr total de ore asistate integral/sem.	49, din care:	ore curs	21	ore seminar/laborator/proiect			28
3.2 Număr total de ore desfășurate on-line asistate integral/sem.	, din care:	ore curs		ore seminar/laborator/proiect			
3.3 Număr de ore asistate parțial/saptămână	, din care:	ore proiect, cercetare		ore practică		ore elaborare lucrare de disertație	
3.3* Număr total de ore asistate parțial/ semestru	, din care:	ore proiect cercetare		ore practică		ore elaborare lucrare de disertație	
3.4 Număr de ore activități neasistate/ săptămână	5.4, din care:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					1.8
		ore studiu individual după manual, suport de curs, bibliografie și notițe					1.8
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri					1.8
3.4* Număr total de ore activități neasistate/ semestru	76, din care:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					25
		ore studiu individual după manual, suport de curs, bibliografie și notițe					25
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri					26
3.5 Total ore/săptămână[9]	8.9						
3.5* Total ore/semestru	125						
3.6 Număr de credite	5						

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	• Programare orientata pe obiecte, Programare web
4.2 de rezultate ale învățării	• Programare orientata pe obiecte (Java, Python), Programare web

5. Condiţii (acolo unde este cazul)

5.1 de desfăşurare a cursului	• Sală curs, Materiale suport: laptop, proiector, tablă
5.2 de desfăşurare a activităţilor practice	• Laborator cu 15-20 calculatoare – Mediu de programare pentru limbajul Java, Python, JavaScript, TypeScript, tablă.

6. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	• C1. Studentul/absolventul alege, descrie, analizează și explică paradigmele moderne de programare, inclusiv programarea funcțională, orientată pe obiect și paralelă, utilizând limbaje și framework-uri actuale. • CC1. Studentul/absolventul evaluează critic limbaje de programare și tehnologii existente, dezvolta sisteme informatice performante și sigure și înțelege importanța proiectării unor sisteme adaptate nevoilor reale ale industriei.
Abilități	• A1.1 Studentul/absolventul proiectează, planifică, construiește, dezvoltă aplicații software complexe ce utilizează eficient resursele hardware și software. • A1.2 . Studentul/absolventul caută, descoperă și înțelege noile tehnologii de dezvoltare a sistemelor informatice pe măsura ce acestea apar. • AC1. Studentul/absolventul dezvoltă și revizuiește proiecte și sisteme informatice, punând accent pe utilitatea si siguranța acestora.
Responsabilitate și autonomie	• RA1.1 Studentul/absolventul produce software și îl adaptează continuu la noile tehnologii și cerințe de piață. • RA1.2 Studentul/absolventul asimilează în procesul de dezvoltare software noi limbaje de programare, tehnologii, biblioteci, cadre de dezvoltare aplicații. • RAC1. Studentul/absolventul manifestă inițiativă și responsabilitate privind realizarea de aplicații complexe, specifice domeniului, inclusiv partea de interacțiune cu utilizatorii, prin prisma înțelegerii nevoilor acestora.

7. Obiectivele disciplinei (asociate rezultatelor învățării specifice acumulate)

- Nevoia aplicatiilor industriale de a evolua pentru a fi capabile sa se adapteze rapid unor noi cerinte este imperativa. Insa, in general, evolutia aplicatiilor industriale este posibila doar daca acestea prezinta atat la nivelul proiectarii cat si al implementarii un anumit standard de calitate. In acest context, cursul propune prezentarea viitorilor absolventi de master in informatica criterii de proiectare, evaluare si refactorizare pentru aplicatii software industriale, in vederea asigurarii unui standard inalt de calitate.
- În cadrul cursului se vor prezenta fazele procesului de evoluţie a software-ului, iar in cadrul fiecărei faze vor fi detaliate activităţile şi instrumentele software ce pot fi folosite pentru automatizarea procesului.

8. Conţinuturi

8.1 Curs	Număr de ore	Din care on-line	Metode de predare
1. Aplicatii industriale 1.1 Terminologie 1.2 Nepotrivita Impedanta Relational-Obiectuala 1.3 Tipare de proiectare in aplicatii industriale 1.4 Conflicte intre tipare de proiectare specifice sistemelor orientate pe obiecte si cele specifice sistemelor enterprise (Data Class vs. Data Transfer Object) 1.5 Tipare de proiectare lingvistice	3	0	Prezentări însoțite de diapozitive. Mostre de cod sursă. Materialele (diapozitive, cod sursă) vor fi disponibile in Campus Virtual (Id Curs = 194) Va fi creată o echipă Microsoft Team și studenților li se va cere să se înroleze – la cerere putem organiza întâlniri 1 la 1 folosind
2. Evaluarea calitatii sistemelor industriale 2.1 Tehnici de Captare a Modelului (Reverse Engineering) 2.2 Detectia Carentelor de Proiectare 2.3 Infrastructura Analizei Automate a Codului (Metodologie; Meta-modelare; Instrumente Software) 2.4 Modele de Calitate	6	0	
3. Refactorizarea sistemelor industriale 3.1 Terminologie 3.2 Introducerea de Tipare de Refactorizare	6	0	
4. Predictia de defecte in aplicatiile industriale folosind invatarea automata (data mining) 4.1 Tehnici de captare a evolutiei aplicatiilor	6	0	

industriale 4.2 Predictia modificarilor si a problemelor bazata pe metrici 4.3 Predictia modificarilor si a problemelor bazata pe structura echipei si a contributiei autorilor			platforma menționatăt

	Bibliografie[10]

8.2 Activităţi aplicative[11]	Număr de ore	Din care on-line	Metode de predare
Pornind de la codul sursa si de la documentatia unei aplicatii industriale de dimensiuni medii se cere studentilor sa analizeze aplicatia sub aspectul calitatii proiectarii codului.	4	0	Expunere temă, discuţii, întrebări fată-în-față sau online prin intermediul platformei Microsoft Teams.
Pe baza cunostintelor dobandite la curs, si folosindu-se de instrumente software adecvate sa se detecteze principalele carente de proiectare dintr-o aplicatie industriala.	4	0	
Sa se reproiecteze la nivel de principiu (si partial la nivel de detaliu, adica modificarea efectiva a codului) aplicatia inspectata.	8	0	
Studiul evolutiei aplicatiilor industriale: 1) Pentru fiecare entitate dintr-o aplicatie extrageti in aceeasi structura de date numarul de revizii si numarul de linii de cod. 2) Pentru fiecare entitate dintr-o aplicatie extrageti in aceeasi structura de date numarul de revizii si numarul de autori. 3) Pentru fiecare entitate dintr-o aplicatie extrageti autorul cu cele mai multe linii de cod introduse.	4	0	
Pe baza cunostintelor dobandite la curs, si folosindu-se de instrumente software adecvate sa se calculeze pentru fiecare entitate din aplicatie diferite metrici software si pe baza valorilor calculate sa se identifice entitatile care vor genera defecte.	6	0	
Recuperari	2	0	

	Bibliografie[12] Bibliografie [aceeasi si pentru Sectiunea 8.1] Programare Java. Petru Florin Mihancea, Cristina Marinescu, Editura Politehnica Timisoara, 2011, ISBN 978-606-554-285-3 Object-Oriented Metrics in Practice: Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems. M. Lanza, R. Marinescu, Springer, 2006. ISBN 978-3540244295 The Art of Readable Code (Theory in Practice). Dustin Boswell, O'Reilly Media, 2011. ISBN 978-0596802295 Software Design X-Rays: Fix Technical Debt with Behavioral Code Analysis. A. Tornhill, Pragmatic Bookshelf, 2018. ISBN 978-1680502725

	Refactoring: Improving the Design of Existing Code. M. Fowler, Addison-Wesley Professional, 2018. ISBN 978-0134757599 Five Lines of Code: How and When to Refactor. C. Clausen, Manning Publications, 2021. ISBN 978-1617298318 Your Code as a Crime Scene, Second Edition: Use Forensic Techniques to Arrest Defects, Bottlenecks, and Bad Design in Your Programs. A. Tornhill, Pragmatic Bookshelf, 2024. ISBN 979-8888650325 https://cv.upt.ro/course/view.php?id=194

9. Evaluare

Tip activitate	**9.1** Criterii de evaluare[13]	**9.2** Metode de evaluare	**9.3** Pondere din nota finală
9.4 Curs	Examinare scrisa	Examen scris / Chestionar online	50%
9.5 Activităţi aplicative	**S:**		
	L: Prezentarea diverselor teme predate la curs Aplicabilitatea practică a noțiunilor însușite	Media aritmetică între notele aferente lucrărilor predate ce vor fi încărcate in campus virtual	50%
	P:		
	Pr:		
	Tc-R[14]:		
9.6 Standard minim de performanţă (volumul de cunoştinţe minim necesar pentru promovarea disciplinei şi modul în care se verifică stăpânirea lui)[15]			
• Evolutia aplicatiilor industriale (terminologie, reguli de evolutie) Stăpânirea a diferite tehnici de refactorizare Calcularea de metrici software Predictia entitatilor problematice			

Data completării	**Titular de curs (semnătura)**	**Titular activităţi aplicative (semnătura)**
21.09.2026	Cristina MARINESCU	Cristina MARINESCU

Director de departament (semnătura)	**Data avizării în Consiliul Facultăţii[16]**	**Decan (semnătura)**
Prof.univ.dr.ing. Mihai MICEA		ș.l.univ.dr.ing. Robert KRISTOF