

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Tehnică din Cluj-Napoca
1.2 Facultatea	Automatică și Calculatoare
1.3 Departamentul	Calculatoare
1.4 Domeniul de studii	Calculatoare și Tehnologia Informației
1.5 Ciclul de studii	Master
1.6 Programul de studii / Calificarea	Complemente de Știința Calculatoarelor
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Tehnici de Programare Fundamentale				Codul disciplinei	9.17
2.2 Titularul de curs	Prof. dr. ing. Cioara Tudor - Tudor.Cioara@cs.utcluj.ro Conf. dr. ing. Chifu Viorica - Viorica.Chifu@cs.utcluj.ro					
2.3 Titularul activităților de seminar / laborator / proiect	Prof. dr. ing. Cioara Tudor - Tudor.Cioara@cs.utcluj.ro Conf. dr. ing. Chifu Viorica - Viorica.Chifu@cs.utcluj.ro					
2.4 Anul de studiu	I	2.5 Semestrul	2	2.6 Tipul de evaluare	E	
2.7 Regimul disciplinei	Categoría formativă: <i>DF – fundamentală, DS - de specialitate, DC - complementară</i>					DF
	Opționalitate: <i>DOB - obligatorie, DOP - opțională, DFA - facultativă</i>					DOP

3. Timpul total estimate

3.1 Număr de ore pe săptămână	4	din care:	3.2 Curs	2	3.3 Seminar	-	3.3 Laborator	2	3.3 Proiect	-
3.4 Număr de ore pe semestru	56	din care:	3.5 Curs	28	3.6 Seminar	-	3.6 Laborator	28	3.6 Proiect	-
3.7 Distribuția fondului de timp (ore pe semestru) pentru:										
(a) Studiul după manual, suport de curs, bibliografie și notițe										10
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										10
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										20
(d) Tutoriat										-
(e) Examinări										4
(f) Alte activități:										-
3.8 Total ore studiu individual (suma (3.7(a)...3.7(f)))					44					
3.9 Total ore pe semestru (3.4+3.8)					100					
3.10 Numărul de credite					4					

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Bazele Programării Orientate pe Obiect
4.2 de competențe	Cunoștințe legate de programare orientată pe obiect

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tabla, proiector, calculator;
5.2. de desfășurare a seminarului / laboratorului / proiectului	Tabla, proiector, calculator, internet, software specific;

6. Competențele specifice acumulate

Competențe profesionale	- analizează specificații software - crează diagrama de proces - dezvoltă prototipul pentru software - identifică cerințele clienților - propune soluții de TIC pentru probleme de afaceri - utilizează șabloane de proiectare de software - utilizează limbaje de interogare - utilizează programarea orientată pe obiect
Competențe transversale	Absolventul: - dezvoltă o abordare analitică - dă dovadă de dorință de învățare - dezvoltă strategii pentru rezolvarea problemelor - este deschis la schimbare - lucrează în echipe

7. Rezultatele așteptate ale învățării

Cunoștințe	Studentul cunoaște: programare pe calculator
Aptitudini	Studentul este capabil să - alinieze software-ul la arhitecturile de sistem - definească cerințele tehnice - evalueze cunoștințele în domeniul TIC - dezvolte prototipul pentru software - utilizeze șabloane de proiectare de software - dezvolte idei creative - identifice cerințele clienților - utilizeze programarea orientată pe obiect
Responsabilitate și autonomie:	Studentul are capacitatea de a lucra independent pentru: - a dezvolta o abordare analitică - a da dovadă de dorință de învățare - a dezvolta strategii pentru rezolvarea problemelor - a fi deschis la schimbare - a lucra eficient în echipe

8. Obiectivele disciplinei (reieșind din grila competențelor specifice acumulate)

8.1 Obiectivul general al disciplinei	Cunoașterea și aplicarea tehnicilor de programare orientate pe obiect în dezvoltarea aplicațiilor software
8.2 Obiectivele specifice	- Sa poata utiliza tehnici de programare pentru proiectarea claselor și interfețelor, incluzând contracte și invariante - Sa poata utiliza tehnici de programare pentru reutilizarea codului folosind mostenirea și polimorfismul - Sa poata utiliza tehnici de programare bazate pe generics și stream-uri pentru procesarea colecțiilor - Sa poata utiliza tehnici de programare pentru reflection, design pattern-uri și framework-uri în vederea reutilizării soluțiilor de proiectare - Sa aplice principiile de proiectare SOLID și thread-urile în Java - Sa poata utiliza tehnici de programare orientate-obiect și functionale într-o abordare integrată pentru dezvoltarea de programe flexibile și eficiente - Sa poata utiliza expresiile lambda și să poată efectua operații de procesare a stream-urilor

9. Conținuturi

9.1 Curs	Nr. ore	Metode de predare	Observații
Introducere - Procesul de dezvoltare software si paradigme de programare	2	Folosirea metodelor multimedia de predare si acces la Internet. Prezentarea cursurilor si discutii. Intrebări adresate studenților în timpul cursului.	N/A
Proiectare - Diagrame UML	2		
Paradigme de programare orientate pe obiect	2		
Tehnici de programare cu thread-uri	2		
Tehnici de programare cu clase abstracte si interfete	2		
Tehnici de compunere si reflectie	2		
Tehnici de design al claselor	2		
Tehnici de programare folosind contracte si invarianti	2		
Principii SOLID, Inversion of Control si framework-uri	2		
Flexibilitate si reutilizare folosind design pattern-uri	4		
Tehnici de programare generice	2		
Expresii lambda si procesarea stream-urilor	4		
Bibliografie: 1. B. Eckel, On Java 8, MindView LLC, 2017 2. E. Gamma, R. Helm, R. Johnson, J. Vlissides - Design Patterns, Addison Wesley Professional, 1994 3. K. Sharan, P. Späth, More Java 17: An In-Depth Exploration of the Java Language and Its Features 3rd Edition, Apress, 2021 4. R. Urma, M. Fusco, A. Mycroft, Modern Java in Action: Lambdas, streams, functional and reactive programming, 2nd Edition, Manning, 2018 5. Note de curs online puse la dispozitie de titularii de curs - Online : http://docs.oracle.com/javase/tutorial/index.html http://stackoverflow.com/			
9.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
Introducere – resurse si cerinte pentru laborator	2	Scurta prezentare a temelor de laborator, discutii pe baza temelor, implementarea temelor pe calculator, discutii si evaluari.	N/A
Tema 1 – Tehnici de Programare cu mostenire si polimorfism	4		
Tema 2 - Tehnici de Programare cu fire de executie	6		
Tema 3 - Tehnici de Programare cu baze de date, design pattern-uri si tehnici de reflectie	6		
Tema 4 - Tehnici de Programare cu Java Collection Framework, expresii lambda si procesare de stream-uri	2		
Evaluare teme laborator	6		
Colocviu de laborator	2		
Bibliografie (bibliografia minimală pentru aplicații conținând cel puțin o lucrare bibliografică de referință a disciplinei care există la dispoziția studenților într-un număr de exemplare corespunzător) - Online : http://docs.oracle.com/javase/tutorial/index.html http://stackoverflow.com/			

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

Este o disciplină a domeniului "Calculatoare și Tehnologia Informației". Ea îi instruieste pe studenți în aplicarea tehnicilor de programare orientate pe obiect în proiectarea și implementarea aplicațiilor software. Conținutul disciplinei a fost stabilit pe baza analizei disciplinelor echivalente de la alte universități precum și pe baza cerințelor angajatorilor IT din România. De asemenea conținutul disciplinei a fost evaluat de agenții guvernamentale românești (CNEAA și ARACIS).

11. Evaluare

Tip activitate	11.1 Criterii de evaluare	11.2 Metode de evaluare (și forma evaluare: continuă/sumativă)	11.3 Pondere din nota finală
11.4 Curs	Cunoasterea și utilizarea tehnicilor de programare prezentate în cadrul cursurilor; prezenta și interacțiunea în cadrul cursurilor.	Examen scris – evaluare sumativa	50%

11.5 Laborator	- Abilitatea de proiectare si implementare a programelor orientate pe obiect. - Abilitatea de a aplica tehnici de programare in practica. - Calitatea codului scris si a diagramelor UML realizate in etapa de proiectare. - Activitatea si prezenta in cadrul sesiunilor de laborator.	Evaluarea temelor de laborator pe parcursul semestrului fata in fata – evaluare continua Colocviu de laborator – evaluare sumativa	50%
11.6 Standard minim de performanță: Sa poata utiliza tehnici de programare orientate pe obiect in proiectarea si implementarea aplicatiilor software. Conditii de promovare a laboratorului: • Predarea la timp a tuturor temelor de laborator. • Minim nota 5 pe fiecare tema. • Prezenta la cel putin 11 lucrari de laborator. • Medie note teme laborator ≥ 5; Nota colocviu ≥ 5 Calcul nota laborator: 50% media notelor pe teme + 50% nota colocviu. Conditii de participare la examenul final: promovarea laboratorului. Calcul nota disciplina: 50% laborator + 50% examen final. Conditii de promovare disciplina: Examen final ≥ 5 Conditii de predare teme de laborator restante: o singura tema poate fi amanata si prezentata intr-una dintre sesiunile de restante.			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
29.04.2026	Curs	Prof. dr. ing. Tudor CIOARA	
		Conf. dr. ing. Viorica CHIFU	
	Aplicații	Prof. dr. ing. Tudor CIOARA	
		Conf. dr. ing. Viorica CHIFU	

Data avizării în Consiliul Departamentului Calculatoare	Director Departament, Prof.dr.ing. Rodica Potolea
Data aprobării în Consiliul Facultății de Automatică și Calculatoare	Decan, Prof.dr.ing. Vlad Mureșan