

FIȘA DISCIPLINEI

1. Date despre program

1.1 Instituția de învățământ superior	UNIVERSITATEA DIN ORADEA
1.2 Facultatea	INGINERIE ELECTRICĂ ȘI TEHNOLOGIA INFORMAȚIEI
1.3 Departamentul	CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
1.4 Domeniul de studii	CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI
1.5 Ciclul de studii	LICENȚĂ
1.6 Programul de studii/Calificarea	CALCULATOARE / INGINER

2. Date despre disciplină

2.1 Denumirea disciplinei	PARADIGME DE PROGRAMARE						
2.2 Titularul activităților de curs	Conf.univ.dr.ing. MIRELA PATER						
2.3 Titularul activităților de seminar /laborator/proiect	Asoc. As. Ing. Remeș Ximena						
2.4 Anul de studiu	II	2.5 Semestrul	3	2.6 Tipul de evaluare	Ex	2.7 Regimul disciplinei	I

(I) Impusă; (O) Opțională; (F) Facultativă

3. Timpul total estimat (ore pe semestru al activităților didactice)

3.1 Număr de ore pe săptămână	4	din care: 3.2 curs	2	3.3 seminar/laborator/proiect	0/2/0
3.4 Total ore din planul de învățământ	56	din care: 3.5 curs	28	3.6 seminar/laborator/proiect	0/28/0
Distribuția fondului de timp					ore
Studiul după manual, suport de curs, bibliografie și notițe					14
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri					28
Tutoriat					7
Examinări					10
Alte activități.....					
3.7 Total ore studiu individual	69				
3.9 Total ore pe semestru	125				
3.10 Numărul de credite	5				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Noțiunile de bază aferente disciplinelor din anul I
4.2 de competențe	

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Cursul se desfășoară față în față. Cursul se desfășoară cu tehnicile disponibile: Laptop, Videoproiector, Tablă inteligentă
5.2. de desfășurare a seminarului/laboratorului/proiectului	Prezența obligatorie la laborator 100% Laboratorul se poate desfășura față în față. Recuperarea laboratoarelor se face conform Regulament privind activitatea profesională a studenților în baza Sistemului European de Credite Transferabile (ECTS) Nivelul maxim admis al absențelor motivate sau nemotivate la activitățile practice (laboratoare, stagii, proiecte etc.) care pot fi recuperate este de până la 30% din totalul activităților. Lucrările de laborator se realizează utilizând mijloacele de lucru existente în laborator:

6.1. Competențele specifice acumulate

Competențe profesionale	C1. Definiște arhitectura software C4. Proiectează sistemul informatic C5. Remediază erorile din software
-------------------------	---

	C6. Utilizează instrumente de inginerie software asistată de calculator
Competențe transversale	CT1. Gândește analitic CT2. Lucrează în echipe
6.2.. Rezultatele învățării	
Cunoștințe	Studentul/absolventul cunoaște conceptele fundamentale ale programării orientate pe obiecte (clase, obiecte, moștenire, polimorfism) și înțelege modul în care acestea se aplică în rezolvarea problemelor informatice prin limbajul Java.
Aptitudini	Studentul/absolventul aplică metode logice și tehnici specifice programării orientate pe obiecte pentru formularea și soluționarea problemelor software, utilizând clase, obiecte, moștenire, polimorfism și supraîncărcarea operatorilor și funcțiilor în limbajul Java..
Responsabilitate și autonomie	Studentul/absolventul demonstrează rigoare științifică și autonomie în dezvoltarea de aplicații Java, implementând și testând programe orientate pe obiecte și utilizând facilitățile obiectuale ale limbajului pentru a rezolva probleme complexe.

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

7.1 Obiectivul general al disciplinei	Cursul are ca scop familiarizarea studenților cu tehnica de programare orientată pe obiecte. Cursul prezintă noțiunile fundamentale de bază ale programării obiectuale cu exemplificări prin programe Java. În cadrul laboratorului studenții implementează și verifică pe calculator atât programele discutate la curs cât și alte programe propuse, aprofundând noțiunile teoretice și practice dobândite. S-a considerat necesară studierea unui limbaj de programare de nivel înalt cu largă răspândire și de actualitate, respectiv limbajul Java.
7.2 Obiectivele specifice	Cunoștințe teoretice: - Utilizarea adecvată în comunicarea profesională a conceptelor proprii calculabilității, complexității, paradigmelor de programare și modelării sistemelor de calcul și comunicații - Utilizarea de teorii și instrumente specifice (algoritmi, scheme, modele, etc.) pentru explicarea funcționării și structurii sistemelor software - Să cunoască conceptele fundamentale ale programării orientate pe obiecte, conceptele de clase și obiecte, constructori și destructori, tehnicile de supraîncărcare a operatorilor și a funcțiilor, tehnica moștenirii și derivării claselor, a polimorfismului - Să cunoască facilitățile obiectuale oferite de limbajul de programare Java Abilități dobândite: - Să stăpânească și să folosească limbajul de programare Java - Să utilizeze în crearea aplicațiilor facilitățile obiectuale oferite de limbajul de programare Java - Să rezolve diverse probleme folosind conceptele de clase, obiecte - Să rezolve diferite probleme folosind tehnicile de supraîncărcare a operatorilor și a funcțiilor, de moștenire și polimorfism

8. Conținuturi*

8.1 Curs	Metode de predare	Nr. Ore / Observații
1.Introducere în paradigme de programare.		2
2. Paradigma structurată		2
3. Paradigma orientată pe obiecte		2

4. Paradigma secvențială versus concurentă.	Videoproiector, slide-uri și predare interactivă la tablă	2
5. Paradigma logică versus funcțională		2
6. Concepte fundamentale în POO – Premisele POO. Concepte fundamentale. Scurta caracterizare a limbajului Java.		2
7. Bazele Java: Object and Driver Classes; Tipuri de date si operatori; Siruri de caractere.		2
8. Declarațiile condiționate; Declarațiile de control		2
9. Siruri de caractere si exceptii.		2
10. Clase, obiecte si metode.		2
11. Parametrii si supraincercarea metodelor.		2
12. Modificatorul static si clase imbricate.		2
13. Moștenire. Polimorfism		2
14. Clase abstracte si generice.		2
Bibliografie 1. Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft, <i>Modern Java in Action</i> , 2nd edition, Manning, 2021 2. Cay S. Horstmann, <i>Core Java Volume I – Fundamentals</i> , 12th edition, Pearson, 2021 3. Herbert Schildt, <i>Java: The Complete Reference</i> , 13th edition, McGraw-Hill, 2023 4. Y. Daniel Liang, <i>Introduction to Java</i> , 11th edition, Pearson, 2021 5. Joshua Bloch, <i>Effective Java</i> , 3rd edition, Addison-Wesley, 2018 (în practică este acceptabilă, dar caută ediția revizuită sau actualizări) 6. Mitsunori Ogihara, <i>Fundamentals of Java Programming</i> , 2020 7. Data Structures & Algorithms in Java (ediție recentă) 8. Florian-Mircea Boian, Rares Florin Boian, <i>Tehnologii fundamentale Java pentru aplicații Web</i> , 2025 9. Venkat Subramaniam – <i>Programming Groovy & Functional Programming with Java</i> (O’Reilly, actualizat 2020)		
8.2 Seminar	Metode de predare	Nr. Ore / Observații
8.3 Laborator		
Introducere. Tehnologii utilizate in cadrul laboratorului: Eclipse	Studiu experimental, programare , dezbateri.	2
Aplicatii clase si obiecte, tipuri de date si operatori, siruri de caractere	Studiu experimental, programare , dezbateri.	2
Aplicatii declaratii	Studiu experimental, programare , dezbateri.	2
Aplicatii siruri de caractere si exceptii	Studiu experimental, programare , dezbateri.	2
Aplicatii clase, obiecte si metode	Studiu experimental, programare , dezbateri.	2
Aplicatii Parametrii si supraincercarea metodelor	Studiu experimental, programare , dezbateri.	2
Aplicatii modificatorul static si clase imbricate	Studiu experimental, programare , dezbateri.	2
Aplicatii mostenire	Studiu experimental, programare , dezbateri.	2
Aplicatii polimorfism	Studiu experimental, programare , dezbateri.	2
Aplicatii interfete	Studiu experimental, programare , dezbateri.	2
Aplicatii clase abstracte si generice	Studiu experimental, programare , dezbateri.	2

Aplicatii colectii	Studiu experimental, programare , dezbatare.	2
Aplicatii sortari si cautari	Studiu experimental, programare , dezbatare.	2
Evaluare finală laborator – Test practic		2
8.4 Proiect		
Bibliografie 1. Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft, <i>Modern Java in Action</i> , 2nd edition, Manning, 2021 2. Cay S. Horstmann, <i>Core Java Volume I – Fundamentals</i> , 12th edition, Pearson, 2021 3. Herbert Schildt, <i>Java: The Complete Reference</i> , 13th edition, McGraw-Hill, 2023 4. Y. Daniel Liang, <i>Introduction to Java</i> , 11th edition, Pearson, 2021 5. Joshua Bloch, <i>Effective Java</i> , 3rd edition, Addison-Wesley, 2018 (în practică este acceptabilă, dar caută ediția revizuită sau actualizări) 6. Mitsunori Ogihara, <i>Fundamentals of Java Programming</i> , 2020 7. Data Structures & Algorithms in Java (ediție recentă) 8. Florian-Mircea Boian, Rares Florin Boian, <i>Tehnologii fundamentale Java pentru aplicații Web</i> , versiune recentă 9. <i>Modern Java in Action</i> — pagina oficială la Manning: https://www.manning.com/books/modern-java-in-action Manning Publications 10. <i>Effective Java, 3rd Edition</i> — pe pagina Addison-Wesley / Pearson: https://www.pearson.com/en-us/subject-catalog/p/effective-java/9780134686042 11. <i>Effective Java</i> (O'Reilly) — versiunea digitală / detalii: https://www.oreilly.com/library/view/effective-java-3rd/9780134686097/		

* Se va detalia conținutul, respectiv numărul de ore alocat fiecărui curs/seminar/laborator/proiect pe durata celor 14 săptămâni ale fiecărui semestru al anului universitar.

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

Conținutul disciplinei se regăsește în curricula specializării de Calculatoare și Tehnologia Informației din alte centre universitare care au acreditate aceste specializări (Universitatea Tehnică din Cluj-Napoca, Universitatea din Craiova, Universitatea „Politehnica” din Timișoara, Universitatea Gh. Asachi Iași, etc), iar cunoașterea principiilor de bază ale programării orientate pe obiecte și implementarea componentelor software, realizarea unor programe pe arii de cunoștințe sunt cerințe stringente ale angajatorilor din domeniu (Qubiz, DecIT, Accessa, Trencadis, Diosoft, Five Tailors, etc).

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoașterea și înțelegerea principiilor programării funcționale, explicarea corectitudinii programelor, aplicarea tehnicilor de raționament formal.	Evaluarea se face față în față, test grilă Examen scris	60%
10.5 Seminar			
10.6 Laborator	- pentru nota 5, cunoașterea lucrărilor de laborator , fără a prezenta detalii asupra acestora - pentru nota 10, cunoașterea amănunțită a modalității de implementare practică a tuturor lucrărilor de laborator	Test practic	40%
Evaluare continuă (40%) și Evaluare sumativă (60%) Se va calcula o notă folosind media aritmetică ponderată ($4 * \text{nota_curs} + 3 * \text{nota_mini-proiect} + 3 * \text{nota_test-laborator}$)/10.			
10.7 Proiect			

10.8 Standard minim de performanță

Selecția și utilizarea independentă a metodelor și algoritmilor învățați pentru situații tip cunoscute precum și finalizarea de calcule.

Dezvoltarea și implementarea algoritmilor prin folosirea principiilor învățate.

Soluționarea la termen, în activități individuale și activități desfășurate în grup, în condiții de asistență calificată, a problemelor care necesită aplicarea de principii și reguli respectând normele deontologiei profesionale.

Obține minim nota 5 atât la curs cât și la laborator.

Prezența la laborator 100%

Data completării

02.09.2025

Semnătura titularului de curs

Conf.univ.dr.ing.Mirela PATER

mpater@uoradea.ro

Semnătura titularului de laborator

Asoc as.ing Remeș Ximena

Data avizării în departament

24.09.2025

Semnătura directorului de departament

Conf. dr. inf. Moisi Elisa

emoisi@uoradea.ro

Data avizării în consiliul facultății

25.09.2025

Semnătură Decan

Conf.dr.ing. Eugen GERGELY

egergely@uoradea.ro