

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 / Master
1.7 Forma de învățământ	IF – învățământ cu frecvență

2. Date despre disciplină

2.1 Denumirea disciplinei	Bazele Programarii OO				Codul disciplinei	6
2.2 Titularul de curs	Conf.dr.ing. Chifu Viorica - Viorica.Chifu@cs.utcluj.ro					
2.3 Titularul activităților de seminar / laborator / proiect	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	Categorii formative: <i>DF – fundamentală, DS - de specialitate, DC - complementară</i>					DF
	Opționalitate: <i>DOB - obligatorie, DOP - opțională, DFA - facultativă</i>					DOB

3. Timpul total estimat

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										20
(b) Documentare suplimentară în bibliotecă, pe platforme electronice de specialitate și pe teren										10
(c) Pregătire seminarii / laboratoare, teme, referate, portofolii și eseuri										10
(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					6					

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Nu este cazul
4.2 de competențe	Cunoașterea limbii engleze la nivel care să permită înțelegerea documentației de specialitate.

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	Tabla, Calculator, videoproiector, platforma Microsoft Teams
5.2. de desfășurare a seminarului / laboratorului / proiectului	Calculatoare, software specific, conexiune la Internet pentru fiecare calculator.

6. Competențele specifice acumulate

Competențe profesionale	• interpretează texte tehnice (documentație, API-uri) • remediază erorile din software • utilizează biblioteci de software (frameworks de bază, clase standard) • utilizează instrumente de inginerie software asistată de calculator • utilizează interfețe specifice aplicațiilor • utilizează șabloane de proiectare de software (pattern-uri simple: Singleton, MVC) • proiectează interfața cu utilizatorul • integrează componente de sistem • se adaptează la schimbările din planurile de dezvoltare tehnologică • utilizează programarea orientată pe obiect • creează modele de date (clase, relații între obiecte) • efectuează testări de sistem (testarea codului scris în laborator)
Competențe transversale	Absolventul: • dezvoltă o abordare analitică • dă dovadă de dorință de învățare • dezvoltă strategii pentru rezolvarea problemelor • lucrează în echipe

7. Rezultatele așteptate ale învățării

Cunoștințe	Studentul cunoaște: • ciclul de viață al dezvoltării sistemelor • programare pe calculator • software pentru mediu integrat de dezvoltare • instrumente de eliminare a erorilor în TIC
Aptitudini	Studentul este capabil să • creeze modele de date • efectueze testări de sistem • integreze componentele de sistem • utilizeze interfețele specifice aplicațiilor • analizeze specificații software • dezvolte prototipul pentru software • interpreteze texte tehnice • remedieze erorile din software • utilizeze biblioteci de software • utilizeze instrumente de inginerie software asistată de calculator • utilizeze șabloane de proiectare de software (pattern-uri elementare) • proiecteze interfața cu utilizatorul • se adapteze la schimbările din planurile de dezvoltare tehnologică • 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 lucra eficient în echipe

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

8.1 Obiectivul general al disciplinei	Studiul aprofundat al conceptelor și paradigmei programării orientate pe obiect, precum și dezvoltarea de aplicații software orientate pe obiect, utilizând principii de modelare, reutilizare și gestionare a datelor.
---------------------------------------	--

8.2 Obiectivele specifice	La finalul disciplinei, studenții vor fi capabili: • să își însușească conceptele și principiile fundamentale ale programării orientate pe obiect, incluzând clase, obiecte, moștenire, polimorfism și încapsulare; • să proiecteze și să implementeze clase și interfețe, utilizând clase abstracte pentru definirea comportamentelor comune și interfețe pentru definirea contractelor de programare; • să aplice compoziția ca alternativă la moștenire, în vederea unei mai bune modularizări și reutilizări a codului; • să înțeleagă și să utilizeze funcționalitățile esențiale ale clasei de bază Object (metode precum equals(), hashCode(), toString()); • să dezvolte interfețe grafice pentru utilizator și să gestioneze evenimentele generate de componentele din interfață; • să dezvolte aplicații utilizând Java Collections Framework pentru manipularea eficientă a datelor; • să utilizeze stream-uri Java pentru lucrul cu fișiere și fluxuri de date; • să aplice tehnici de serializare și deserializare a obiectelor; • să efectueze analiza lexicală a fluxurilor de date folosind clase precum StreamTokenizer și StringTokenizer pentru procesarea și interpretarea textelor; • să identifice și să utilizeze șabloane de proiectare (ex. MVC) de bază; • să scrie cod clar, bine structurat și ușor de întreținut, aplicând în mod consecvent principiile programării orientate pe obiect.
---------------------------	---

9. Conținuturi

9.1 Curs	Nr. ore	Metode de predare	Observații
Concepte și paradigme ale programării orientate pe obiect	2	Prezentări susținute cu ajutorul videoproiectorului și al tablei, completate cu expuneri interactive și discuții. Materialele didactice sunt puse la dispoziție pe platforma Microsoft Teams.	
Abstracții și tipuri de date abstracte	2		
Structuri de control în Java	2		
Clase și obiecte. Tablouri	2		
Pachete și organizarea codului. Interfețe Java	2		
Moștenire și polimorfism. Clasa Object	2		
Clase interne (nested/inner classes)	2		
Compoziție și reutilizarea claselor	2		
Erori și excepții în Java	2		
Dezvoltarea interfețelor grafice (Swing) – Partea I	2		
dezvoltarea interfețelor grafice (Swing) – Partea II	2		
Componente avansate pentru GUI în Swing	2		
Java Collections Framework – List, Set și Map	2		
Stream-uri în Java și procesarea datelor			
Bibliografie: 1. Kathy Sierra, Bert Bates, Trisha Gee – <i>Head First Java</i> , Third Edition, O’Reilly Media, 2022. 2. Herbert Schildt – <i>Java: The Complete Reference</i> , Twelfth Edition, McGraw-Hill, 2021. 3. Bruce Eckel – <i>Thinking in Java</i> , Fourth Edition, Prentice Hall, 2006 (disponibil gratuit online). 4. Oracle – <i>The Java™ Tutorials and Java Language Specification</i> (docs.oracle.com). 5. Simon Kenda, <i>Object Oriented Programming Using Java</i>			
9.2 Seminar / laborator / proiect	Nr. ore	Metode de predare	Observații
Familiarizarea cu mediul de dezvoltare IntelliJ IDEA Configurarea proiectelor, structurarea pachetelor și utilizarea instrumentelor de depanare.	2	Scurta prezentare a temelor de laborator, discutii pe baza temelor, implementarea temelor pe calculator,	
Tema 1: Tipuri primitive în Java, variabile și expresii. Declararea și inițializarea variabilelor, conversii de tip, operatori aritmetici și logici	2		
Tema 2: Controlul fluxului și clase în Java. Instrucțiuni de selecție și iterație, definirea și instanțierea claselor, metode, constructori.	3		

Tema 3: Interfețe Java. Definirea și implementarea interfețelor, relația cu clasele abstracte.	3		
Tema 4: Tehnici de reutilizare a claselor – moștenire și polimorfism în Java. Crearea ierarhiilor de clase, utilizarea polimorfismului, suprascrierea metodelor esențiale din clasa Object (toString(), equals(), hashCode()).	4		
Tema 5: Dezvoltarea interfețelor grafice pentru utilizator in Swing. Crearea componentelor vizuale, gestionarea evenimentelor	5		
Tema 6: Lucru cu colecții în Java. Manipularea structurilor de tip List, Set și Map: adăugare, căutare, sortare, iterație.	4		
Tema 7: Lucru cu fluxuri de date (stream-uri) în Java. Citirea și scrierea fișierelor text și binare, serializarea și deserializarea obiectelor	3		
Tema 8: Susținerea colocviului final. Evaluare a cunoștințelor și a competențelor dobândite pe parcursul activităților de laborator.	2		
Bibliografie 1. Tutorial Point - Java Tutorial - https://www.tutorialspoint.com/java/index.htm 2. W3Schools – Java Tutorial - https://www.w3schools.com/java/ 3. Herbert Schildt – Java: A Beginner’s Guide, Eighth Edition , McGraw-Hill, 2019 - https://www.sietk.org/downloads/javabook.pdf 4. Y. Daniel Liang – Introduction to Java Programming and Data Structures, 12th Edition , Pearson, 2020- https://api.pageplace.de/preview/DT0400.9781292402130_A42567323/preview-9781292402130_A42567323.pdf			

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

--

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	Verificarea cunoștințelor teoretice dobândite la curs prin intermediul unui test grila	Examen scris - evaluare sumativa	40%
11.5 Seminar/Laborator /Proiect	Rezolvarea corectă și completă a exercițiilor practice pe parcursul semestrului și susținerea colocviului final	Evaluare continuă a temelor și a colocviului	60%
11.6 Standard minim de performanță: Nota minimă 5 la examenul și la laborator			

Data completării:	Titulari	Titlu Prenume NUME	Semnătura
29.04.2026	Curs	Conf.dr.ing. Viorica-Rozina CHIFU	
	Aplicații	Conf.dr.ing. Viorica-Rozina 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