



FIȘA DISCIPLINEI
PROGRAMARE ORIENTATA PE OBIECTE

1. Date despre program

1.1 Instituția de învățământ superior	UNIVERSITATEA „OVIDIUS” DIN CONSTANȚA
1.2 Facultatea	Matematică și Informatică
1.3 Departamentul	Matematică și Informatică
1.4 Domeniul de studii	Informatică
1.5 Ciclu de studii	Licență
1.6 Programul de studii	Informatică
1.7 Anul universitar	2025-2026

2. Date despre disciplină

2.1 Denumirea disciplinei	Programare orientata pe obiecte						
2.2 Cod disciplină	Info.1.2.17						
2.3 Titularul activităților de curs	Conf.dr. Puchianu Crenguța						
2.4 Titularul activităților aplicative	Conf.dr. Puchianu Crenguța						
2.5 Anul de studii	1	2.6 Semestrul	2	2.7 Tipul de evaluare	Ex	2.8 Regimul disciplinei */**	DS/DOB

* DF – disciplină fundamentală, DS – disciplină de specializare, DC – disciplină complementară

** DOB – disciplină obligatorie; DOP – disciplină opțională; DFA – Disciplină facultativă

3. Timpul total (ore pe semestru)

3.1 Număr de ore activități directe pe săptămână	4	din care: 3.2 curs	2	3.3 aplicații***	2
3.4 Total ore activități directe pe semestru	56	din care: 3.5 curs	28	3.6 aplicații	28
3.7 Total ore de studiu individual					69
Distribuția fondului de timp					[ore]
Studiul cărților, manualelor, suportului de curs, notițelor, bibliografie minimală recomandată					10
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren					13
Pregătire seminar / laborator / proiect, teme, referate, portofolii și eseuri					12
Pregătire pentru prezentări sau verificări					10
Pregătire pentru examinarea finală					14
Alte activități: consultații					10
3.8 Total ore pe semestru	125				
3.9 Numărul de credite	5				

*** S - seminar; L - laborator; P - proiect

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	Fundamentele programarii Algoritmi fundamentali
4.2 de rezultate ale învățării	-



5. Condiții necesare pentru desfășurarea optimă a activităților didactice (acolo unde este cazul)

5.1. de desfășurare a cursului	Sala de curs disponibilă
5.2. de desfășurare a laboratorului	Sala de laborator disponibilă

6. Obiectivele disciplinei

6.1 Obiectivul general al disciplinei	Introduce principiile și mecanismele programării orientate spre obiecte: abstractizare, încapsulare, modularitate și ierarhizare.
6.2 Obiectivele specifice	Însușirea cunoștințelor de bază ale programării orientată spre obiecte și exemplificarea lor folosind limbajul Java.

7. Rezultatele învățării

Cunoștințe	Studentul / Absolventul: <ul style="list-style-type: none">- identifică conceptele domeniului analizat și le transformă în clase sau atribute, pe care apoi le implementează folosind limbajul Java- știe să aplice principiile și mecanismele programării orientată spre obiecte pentru realizarea programelor- cunoaște mai multe metode de realizare a programelor, folosind biblioteca Java
Aptitudini	Studentul / Absolventul: <ul style="list-style-type: none">- aplică paradigma orientată spre obiecte pentru implementarea programelor care rezolvă probleme reale de complexitate medie- stabilește structurile de date necesare implementării algoritmilor utilizați în programe- folosește instrumente informatice specializate în modelarea, implementarea și testarea programelor
Responsabilitate și autonomie	Studentul / Absolventul: <ul style="list-style-type: none">- are o atitudine de discernământ în utilizarea surselor de informare- propune mai multe soluții de implementare a programelor orientate spre obiecte- îndeplinește la termen, în mod riguros, sarcini profesionale cu grad mediu de complexitate

8. Conținuturi

8.1 Curs	Metode de predare	Număr ore alocate
Elemente de baza ale limbajului Java: variabile, expresii, instrucțiuni.	Metode de predare-învățare interactive	2 ore
Conversii de date primitive		2 ore
Variabile și metode statice. Metode supraincarcate. Metode recursive		2 ore
Concepte și instanțe de concepte. Obiecte. Constructori. Supraincarcarea constructorilor. Modificatori de acces. Variabile și metode instanță	Dialogul	2 ore
Tablouri de date. Prelucrarea caracterelor și a sirurilor de caractere.		2 ore
Moștenirea în Java. Relația de generalizare/specializare conceptuală.	Problematizarea	2 ore
Moștenirea simplă în Java. Vizibilitatea membrilor superclasei în subclase.		2 ore
Ierarhii de moștenire. Clasa Object. Constructori și moștenire	Metode active și interactive	2 ore
Polimorfism static și dinamic. Clase abstracte.		2 ore
Interfețe. Utilizarea interfetelor pentru implementarea rolurilor. Moștenire multiplă		2 ore
Gestiunea excepțiilor.		1 ora
Programarea intrărilor și ieșirilor. Clase de flux de octeți de intrare și ieșire.		2 ore
Clase de flux de caractere de intrare și de ieșire		
Interfețe grafice utilizator. Componente grafice. Tipuri de layout-uri: FlowLayout, BorderLayout, GridLayout.		2 ore



UOC-PO-10 Anexa 3

Clase interne și anonime. Definirea claselor interne. Clase interne statice. Clase interne și mostenirea. Clase anonime. Utilizarea claselor interne și anonime în aplicații orientate spre obiecte.	Sintetizarea/esențializarea informațiilor	1 ora
Modelul gestiunii evenimentelor. Evenimente semantice. Evenimente de nivel coborat. Interfete de ascultare. Clase adapter ale interfetelor de ascultare. Interfete functionale. Functii lambda.	Generalizarea	3 ore
Colecții de obiecte. Framework-ul de colectii din biblioteca Java: interfetele Collection, Iterator, Set, clasele HashSet și TreeSet, interfele List și ListIterator, clasele ArrayList și LinkedList, interfata Map, clasele HashMap și TreeMap.	Explicația	2 ore
Tipuri enumerate. Clasa Enum. Folosirea tipurilor enumerate		1 ora
Fluxuri de elemente. Surse pentru fluxuri. Operații intermediare și finale pe fluxuri		2 ore
Bibliografie		
[1]. C. M. Bogdan și L. D. Șerbănași, <i>Dezvoltarea orientată spre obiecte a programelor în Java</i> , Editura Politehnica Press, 2011		
[2]. S. Andrei și colectiv, <i>Java de la 0 la expert</i> . Editura Polirom, 2003		
[3]. Athanasiu și colectiv, <i>Limbaajul Java. O perspectivă pragmatică</i> . Editura Teora, 1998.		
[4]. B. Eckel, <i>Thinking in Java</i> . Prentice Hall, 1998.		
[5]. C. S. Horstmann, <i>Computing Concepts with Java 2 Essentials</i> , Second Edition. John Wiley&Sons, 2000		
8.2 Aplicații (laborator)	Metode de predare	Număr ore alocate
Elemente de baza ale limbajului Java: variabile, expresii, instrucțiuni.	Metode de învățare interactive	2 ore
Conversii de date primitive		2 ore
Variabile și metode statice. Metode supraincarcate. Metode recursive		2 ore
Concepte și instanțe de concepte. Obiecte. Constructori. Supraincarea constructorilor. Modificatori de acces. Variabile și metode instanță		2 ore
Tablouri de date. Prelucrarea caracterelor și a sirurilor de caractere.		2 ore
Moștenirea în Java. Relația de generalizare/specializare conceptuală.		2 ore
Moștenirea simplă în Java. Vizibilitatea membrilor superclaselor în subclase.		
Ierarhii de moștenire. Clasa Object. Constructori și moștenire		
Polimorfism static și dinamic. Clase abstracte.		2 ore
Interfete. Utilizarea interfetelor pentru implementarea rolurilor. Moștenire multiplă		2 ore
Gestiunea excepțiilor.	Metode active și interactive	1 ora
Programarea intrărilor și ieșirilor. Clase de flux de octeți de intrare și ieșire. Clase de flux de caractere de intrare și de ieșire		2 ore
Interfete grafice utilizator. Componente grafice. Tipuri de layout-uri: FlowLayout, BorderLayout, GridLayout.		2 ore
Clase interne și anonime. Definirea claselor interne. Clase interne statice. Clase interne și moștenirea. Clase anonime. Utilizarea claselor interne și anonime în aplicații orientate spre obiecte.	Sintetizarea/esențializarea informațiilor	1 ora
Modelul gestiunii evenimentelor. Evenimente semantice. Evenimente de nivel coborat. Interfete de ascultare. Clase adapter ale interfetelor de ascultare. Interfete functionale. Functii lambda.	Generalizarea	3 ore
Colecții de obiecte. Framework-ul de colectii din biblioteca Java: interfetele Collection, Iterator, Set, clasele HashSet și TreeSet, interfele List și ListIterator, clasele ArrayList și LinkedList, interfata Map, clasele HashMap și TreeMap.	Explicația	2 ore
Tipuri enumerate. Clasa Enum. Folosirea tipurilor enumerate		1 ora
Fluxuri de elemente. Surse pentru fluxuri. Operații intermediare și finale pe fluxuri		2 ore



Bibliografie

- [1]. C. M. Bogdan și L. D. Șerbănați, *Dezvoltarea orientată spre obiecte a programelor în Java*, Editura Politehnica Press, 2011
- [2]. S. Andrei și colectiv, *Java de la 0 la expert*. Editura Polirom, 2003
- [3]. Athanasios și colectiv, *Limbaajul Java. O perspectivă pragmatică*. Editura Teora, 1998.
- [4]. B. Eckel, *Thinking in Java*. Prentice Hall, 1998.
- [5]. C. S. Horstmann, *Computing Concepts with Java 2 Essentials*, Second Edition. John Wiley&Sons, 2000

9. Evaluare

Tip activitate	9.1 Criterii de evaluare	9.2 Metode de evaluare	9.3 Pondere din nota finală
9.4 Curs	Participare activă la activități didactice	Evaluare orală	10%
9.5 Laborator	Completarea a 2 teste grila	Evaluare scrisă/online	30%
	Examinare	Notă examinare	50%
Din oficiu			10%
9.6 Standard minim de performanță / Condiții de promovare			
Identificarea componentelor grafice și a containerelor unei interfețe grafice utilizator. Construirea unui program în Java care să afișeze interfețe grafice utilizator			

Data completării,

Titular activităților de curs,
Conf. univ. dr. Crenguta Puchianu

Titular aplicații,
Conf. univ. dr. Crenguta Puchianu

16.09.2025

Data avizării în Departament,

Director de Departament,
Conf. univ. dr. Elena Pelican

19.09.2025

Decan,
Conf. univ. dr. Aurelian Nicola