

FIȘA DISCIPLINEI¹⁾

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiești
1.2. Facultatea	Litere și Științe
1.3. Departamentul	Informatică, Tehnologia Informației, Matematică și Fizică
1.4. Domeniul de studii universitare	Informatică
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Informatică

2. Date despre disciplină

2.1. Denumirea disciplinei	Programare orientată pe obiecte
2.2. Titularul activităților de curs	Șef lucr. dr. Ioniță Irina
2.3. Titularul activităților seminar/laborator	Șef lucr. dr. Ioniță Irina Lector dr. Dragomir Elia Georgiana
2.4. Titularul activității proiect	-
2.5. Anul de studiu	1
2.6. Semestrul *	2
2.7. Tipul de evaluare	E
2.8. Categoria formativă** / regimul*** disciplinei	DS/O

* numărul semestrului este conform planului de învățământ;

** DF - Discipline fundamentale; DD - discipline de domeniu; DS - discipline de specialitate; DC - discipline complementare, DA - disciplina de aprofundare, DSI - disciplina de sinteză.

*** obligatorie = O; opțională = A; facultativă = L

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	2	3.4. Proiect	-
3.5. Total ore din planul de învățământ	56	din care: 3.6. curs	28	3.7. Seminar/laborator	28	3.8. Proiect	-
3.9. Distribuția fondului de timp							ore
Studiu după manual, suport de curs, bibliografie și notițe							30
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren							10
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri							40
Tutoriat							6
Examinări							8
Alte activități							
3.10 Total ore studiu individual	94						
3.11. Total ore pe semestru	180						
3.12. Numărul de credite	6						

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none">➤ Fundamentele programării➤ Algoritmi fundamentali
4.2. de competențe	<ul style="list-style-type: none">➤ Cunoașterea unui limbaj de programare (limbajul C) și a etapelor de dezvoltare a aplicațiilor cu acest limbaj de programare

¹⁾ Adaptare după Ordinul Ministrului educației, cercetării, tineretului și sportului nr. 5 703/2011 privind implementarea Codului național al calificărilor din învățământul superior, publicat în Monitorul Oficial al României, partea I, nr.880 bis / 13.XII.2011

5. Condiții (acolo unde este cazul)

5.1. de desfășurare a cursului	➤ Cursul se desfășoară într-o sală dotată cu calculator, videoproiector, acces Internet, tablă.
5.2. de desfășurare a seminarului/laboratorului	➤ Laboratorul se desfășoară în săli de laborator, aplicațiile fiind implementate pe stații de lucru într-un mediu de programare pentru limbajul Java (NetBeans).

6. Competențe specifice acumulate

Competențe profesionale	C.1.1. Descrierea adecvată a paradigmei de programare orientată pe obiecte și a mecanismelor de limbaj specifice, precum și identificarea diferenței dintre aspectele de ordin semantic și sintactic C1.2 Explicarea unor aplicații soft existente, pe niveluri de abstractizare (arhitectură, pachete, clase, metode) utilizând în mod adecvat cunoștințele de bază C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente în limbajul Java, pe baza unor specificații C1.4 Testarea unor aplicații orientate-obiect pe baza unor planuri de test. C1.5 Dezvoltarea unor aplicații orientate obiect și elaborarea documentațiilor aferente.
Competențe transversale	CT1. Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic-științific, pentru valorificarea creativă a propriului potențial, cu respectarea principiilor și a normelor de etică profesională CT2. Desfășurarea eficientă a activităților organizate într-un grup inter-disciplinar și dezvoltarea capacităților empatică de comunicare inter-personală, de relaționare și colaborare cu grupuri diverse.

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

7.1. Obiectivul general al disciplinei	➤ Formarea de competențe profesionale și transversale necesare obținerii calificării. Obiectivul principal al disciplinei constă în familiarizarea studenților cu abordarea orientată-obiect și deprinderea acestora cu un limbaj de programare de nivel înalt (Java) pentru dezvoltarea de aplicații cu aplicabilitate în domeniul de specialitate.
7.2. Obiectivele specifice	Formarea următoarelor competențelor profesionale (C1.1 –C1.5) și transversale (CT1, CT2). La sfârșitul cursului, studenții vor fi capabili să: ➤ descrie adecvat paradigma programării orientate pe obiecte prin definirea conceptele de bază și să realizeze o comparație cu celelalte tipuri de programări (ex. programare procedurală); ➤ explice nevoia utilizării programării orientate pe obiecte, identificând avantajele acesteia; ➤ deprindă un nou mod de programare (orientat-obiect) și să învețe un limbaj orientat-obiect (Java); ➤ proiecteze și să implementeze aplicații în limbajul orientat obiect studiat (Java) cu aplicabilitate în domeniul de specialitate; ➤ utilizeze UML pentru analiza și proiectarea aplicațiilor orientate-

	<p>obiect;</p> <p>➤ aplice reguli de muncă organizată și eficientă, să manifeste atitudini responsabile față de domeniul didactic-științific, să respecte principii și norme de etică profesională, să lucreze în echipă la elaborarea de proiecte de complexitate medie.</p>
--	---

8. Conținuturi

8.1. Curs	Nr.ore	Metode de predare	Observații
Nevoia utilizării programării orientate obiect în dezvoltarea aplicațiilor software. Analiza și proiectarea orientată obiect a sistemelor software	2	Prelegere, susținută de prezentări PPT, conversații, explicații, exemplificări practice la tablă, studii de caz	
Paradigma de programare OO vs. Paradigma de programare imperativă	2		
Concepte de bază ale programării orientate pe obiecte (în Java). Tipuri de date abstracte. Încapsularea și ascunderea informațiilor	2		
Obiecte și clase: date și funcții membru	2		
Crearea și distrugerea obiectelor: funcțiile constructor și destructor (Garbage Collector). Tipuri de constructori	2		
Funcții și clase friend. Funcții inline. Membri statici. Tablouri de obiecte	2		
Supraîncărcarea în programarea orientată pe obiecte	2		
Clase imbricate (nested class) Clase template	2		
Moștenirea în programarea orientată pe obiecte	2		
Polimorfismul în programarea orientată pe obiecte	2		
Proiectare ierarhii de clase, interfețe în Java	2		
Design patterns	2		
Diagrame UML – reprezentarea claselor	2		
Dezvoltarea unei aplicații complexe din lumea reală	2		
<p>Bibliografie</p> <p>T. Budd (1997), An Introduction to Object-Oriented Programming, Addison Wesley.</p> <p>S. Guțu, L. Dumitrașcu, G. Marcu, L. Ioniță (2005), Analiza și proiectarea orientată obiect a sistemelor informatice cu UML, INDE Ploiești, Editura Universității din Ploiești, ISBN 973-719-012-2.</p> <p>L. Ioniță, et al (2015), Diagramele UML 2. Dicționar. Studii de caz. Aplicație Web, Editura Universității Petrol-Gaze din Ploiești, Ploiești.</p> <p>M. Preda, A. Mircea, D. Preda, C. Teodorescu (2010), Introducere în programarea orientată-obiect. Concepte</p>			

fundamentale din perspectiva ingineriei software, Editura Polirom, București.

D. Schiopu, I. Ionita, L. Ionita, G. Marcu, L. Dumitrascu (coordonator) (2016), De la C la C++, C#. Exerciții didactice, Editura Universității Petrol-Gaze din Ploiești, Ploiești.

C. Spircu, I. Lopătan (1995), Analiza, proiectarea și programarea orientate spre obiecte, Editura Teora, București.

T. Radulescu (2000), Ingineria software orientată pe obiecte, Editura Matrix Rom, București, 2000.

J. Gosling, B. Joy, G. Steele, G.Bracha, A. Buckley. The Java™ Language Specification Java SE 7 Edition.

B. Eckel(2006), Thinking in Java, 4th edition, Prentice Hall

B. Eckel (2004), Thinking in Patterns with Java, MindView, Inc.

C. Frăsineanu, Curs practic de Java, https://profs.info.uaic.ro/~acf/java/Cristian_Frasinaru-Curs_practic_de_Java.pdf.

<https://javatutorial.net/java-oop>

<https://www.w3resource.com/java-tutorial/java-object-oriented-programming.php>

<https://www.cs.usfca.edu/~parrt/doc/java/OOProgWithJava-notes.pdf>

https://www.ntu.edu.sg/home/ehchua/programming/java/J3a_OOPBasics.html

8.2. Seminar / laborator	Nr. ore	Metode de predare	Observații
Prezentarea mediului de programare NetBeans.	2	Prelegere, discuții, rezolvarea la calculator a unor probleme propuse și explicații la tablă	
Clase și obiecte. Constructori, destructor. Tipuri de constructori	4		
Membri statici. Tablouri de obiecte	2		
Supraîncărcarea în programarea orientată pe obiecte	2		
Clase imbricate. Clase template	2		
Moștenirea în programarea orientată pe obiecte. Ierarhii de clase	4		
Polimorfismul în programarea orientată pe obiecte. Clase abstracte	4		
Interfețe în Java. Design patterns	2		
Diagrame UML pentru reprezentarea claselor. Exemple	2		
Dezvoltarea unei aplicații complexe din lumea reală	4		

Bibliografie			
D. Schiopu, I. Ionita, L. Ionita, G. Marcu, L. Dumitrascu (coordonator) (2016), De la C la C++, C#. Exerciții didactice, Editura Universității Petrol-Gaze din Ploiești, Ploiești.			
C. Spircu, I. Lopătan (1995), Analiza, proiectarea și programarea orientate spre obiecte, Editura Teora, Bucuresti.			
C. Frăsineanu, Curs practic de Java, https://profs.info.uaic.ro/~acf/java/Cristian_Frasinaru-Curs_practic_de_Java.pdf .			
https://javatutorial.net/java-oop			
https://www.w3resource.com/java-tutorial/java-object-oriented-programming.php			
https://www.cs.usfca.edu/~parrt/doc/java/OOProgWithJava-notes.pdf			
https://www.ntu.edu.sg/home/ehchua/programming/java/J3a_OOPBasics.html			
8.3. Proiect	Nr. ore	Metode de predare	Observații

Bibliografie			

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

<ul style="list-style-type: none"> ➤ Cursul și laboratorul sunt astfel concepute încât, prin competențele formate, să răspundă cerințelor pieței muncii. ➤ Disciplina Programare orientată pe obiecte respectă recomandările IEEE și ACM legate de Curricula pentru specializarea Informatică. Noțiunile teoretice prezentate în curs și laborator sunt conform standardelor Internet Engineering Task Force. ➤ Conținuturile disciplinei sunt specifice domeniului dezvoltării aplicațiilor software orientate-obiect, temele de lucru individual și proiectele, precum și lista de probleme aferentă fiecărui capitol din curs sunt astfel alese încât să corespundă tipurilor de activitate specifice specializării.
--

10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Dobândirea cunoștințelor de bază privind programarea orientată pe obiecte	Test grilă	30%
	Însușirea unui limbaj de specialitate (Java) conform domeniului		
10.5. Seminar/laborator	Însușirea tehnicilor de lucru în dezvoltarea de aplicații orientate-obiect specifice domeniului	Teste parțiale de verificare a cunoștințelor acumulate și a stilului de programare ce constau în rezolvarea unor probleme pe calculator	30%
		Examinarea proiectelor realizate de echipe formate din maxim trei studenți, având o temă la alegere, referitoare la dezvoltarea unei aplicații	30%
	Participarea activă la orele de laborator	Întrebări pe parcursul semestrului	10%
10.6. Proiect			
10.7. Standard minim de performanță			
<ul style="list-style-type: none"> ➤ declararea unei clase; declararea și definirea a două funcții membru (un constructor și o funcție la alegere); declararea a două obiecte ce aparțin clasei respective și apelarea unei funcții. ➤ obținerea notei minim 5 la toate cele trei metode de evaluare (test grilă, teste parțiale și proiect). 			

Data completării	Semnătura titularului de curs	Semnătura titularului de seminar/laborator	Semnătura titularului de proiect
14.09.2020	_____	_____	_____

Data avizării în departament	Director de departament <i>(funcție didactică, nume, prenume)</i> <i>(Semnătură)</i>	Decan <i>(funcție didactică, nume, prenume)</i> <i>(Semnătură)</i>
21.09.2020	_____	_____