

FIȘA DISCIPLINEI

1. Date despre program

1.1. Instituția de învățământ superior	Universitatea Petrol-Gaze din Ploiești
1.2. Facultatea	Inginerie Mecanică și Electrică
1.3. Departamentul	Automatică, Calculatoare și Electronică
1.4. Domeniul de studii universitare	Calculatoare și Tehnologia Informației
1.5. Ciclul de studii universitare	Licență
1.6. Programul de studii universitare	Calculatoare

2. Date despre disciplină

2.1. Denumirea disciplinei	Paradigme de programare
2.2. Titularul activităților de curs	Prof. dr. ing. Mihaela Oprea
2.3. Titularul activităților seminar/laborator	Prof. dr. ing. Mihaela Oprea
2.4. Titularul activității proiect	
2.5. Anul de studiu	II
2.6. Semestrul *	4
2.7. Tipul de evaluare	Verificare
2.8. Categoria formativă** / regimul*** disciplinei	DL / DFA

* numărul semestrului este conform planului de învățământ;

** DF - Discipline fundamentale; DS - discipline de specializare; DC - discipline complementare

*** obligatorie/impusă = DOB; opțională = DOP; facultativă = DFA

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

3.1. Număr de ore pe săptămână	3	din care: 3.2. curs	2	3.3. Seminar/laborator	1	3.4. Proiect	
3.5. Total ore din planul de învățământ	42	din care: 3.6. curs	28	3.7. Seminar/laborator	14	3.8. Proiect	
3.9. Total ore studiu individual (studiu după suport de curs, bibliografie și notițe, documentare suplimentară în bibliotecă, pe platformele electronice de specialitate, pregătire seminar/laboratoare, teme, referate, portofolii și eseuri)							8
3.10. Total ore pe semestru							50
3.11. Numărul de credite							2

4. Condiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none">➤ Programarea calculatoarelor și limbaje de programare➤ Programare orientată pe obiecte
4.2. de desfășurare a cursului	<ul style="list-style-type: none">➤ În funcție de modalitatea de susținere a orelor de curs (față în față/online), sală de curs/Google Meet și platforma e-learning a facultății.
4.3. de desfășurare a seminarului/laboratorului	<ul style="list-style-type: none">➤ Laborator dotat cu tehnică de calcul și medii de programare specifice diferitelor paradigme de programare studiate (Java, C++, Prolog, Haskell etc) - în cazul orelor susținute față în față. Platforma e-learning a facultății și Google Meet (în cazul orelor susținute online).

5. Competențe specifice acumulate și rezultatele învățării* care stau la baza acestora

Competențe profesionale	Rezultatele învățării*
-------------------------	------------------------

1. Operarea cu fundamente științifice, ingineresti și ale tehnologiei informației.	<p>C1 - Studentul/absolventul identifică și descrie concepte, principii și metode de bază pentru diferite paradigme de programare (procedurală, declarativă, orientată pe obiecte etc).</p> <p>A1 - Studentul/absolventul analizează sistemele utilizând teoriile studiate și proiectează, implementează, diagnostichează și depanează programe implementate conform paradigmei de programare utilizată.</p> <p>RA1 - Studentul/absolventul selectează și utilizează surse bibliografice specifice domeniului.</p>
2. Soluționarea problemelor folosind instrumentele științei și ingineriei calculatoarelor	<p>C1 - Studentul/absolventul descrie, identifică, sumarizează concepte și metode elementare privitoare la limbaje de programare corespunzătoare diferitelor paradigme de programare, medii de programare, tehnici de programare și inginerie software specifică paradigmei de programare utilizată și modul lor de aplicare în probleme concrete.</p> <p>A1 - Studentul/absolventul alege și explică concepte proprii specifice paradigmei de programare studiate.</p> <p>RA1 - Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.</p> <p>RA2 - Studentul/absolventul selectează și utilizează surse bibliografice specifice domeniului.</p>
3. Proiectarea, gestionarea ciclului de viață și integrarea sistemelor informatice utilizând tehnologii și medii de programare	<p>C1 - Studentul/absolventul identifică, descrie și sumarizează concepte și metode elementare privitoare la limbaje de programare corespunzătoare diferitelor paradigme de programare, medii de programare, tehnici de programare și inginerie software specifică paradigmei de programare studiate și modul lor de aplicare.</p> <p>A1 - Studentul/absolventul elaborează specificații și proiectează sisteme informatice folosind metode și instrumente specifice paradigmei de programare studiate.</p> <p>RA1 - Studentul/absolventul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.</p> <p>RA2 - Studentul/absolventul selectează și utilizează surse bibliografice specifice domeniului.</p>
Competențe transversale	Rezultatele învățării*
1. Aplicarea, în mod responsabil, a principiilor, normelor și valorilor eticii profesionale în realizarea sarcinilor profesionale și identificarea obiectivelor de realizat, a resurselor disponibile, a etapelor de lucru, a duratelor de execuție, a termenelor de realizare aferente și a riscurilor aferente.	<p>C1 - Studentul/absolventul descrie, identifică și sumarizează concepte fundamentale paradigmei de programare studiate și modul lor de aplicare în probleme concrete.</p> <p>A1 - Studentul/absolventul specifică cerințe, elaborează scenarii de simulare, propune soluții de rezolvare a unor probleme de control (folosind diferite paradigme de programare), analizează și evaluează performanțele sistemelor informatice dezvoltate.</p> <p>RA1 - Studentul/absolventul are o comportare onorabilă, responsabilă, etică, în spiritul legii pentru a asigura reputația profesiei.</p> <p>RA2 - Studentul/absolventul aplică valorile eticii și deontologiei profesiei de inginer.</p>
2. Identificarea oportunităților de formare continuă și utilizarea eficientă pentru propria dezvoltare a surselor informaționale și a resurselor de comunicare și formare profesională asistată de calculator (portaluri Internet, aplicații software de specialitate, baze de date, cursuri on-line) atât în limba română, cât și într-o limbă de circulație internațională.	<p>C1 - Studentul/absolventul explică și interpretează rezultate teoretice și experimentale, documentație tehnică, fenomene și procese din domeniul calculatoare și tehnologia informației – pentru fiecare paradigmă de programare (procedurală, neprocedurală).</p> <p>A1 - Studentul/absolventul realizează responsabil proiecte pentru rezolvarea unor probleme specifice domeniului, cu evaluarea corectă a volumului de lucru, a resurselor disponibile, a timpului necesar de finalizare și a riscurilor, în condiții de aplicare a normelor deontologice și de etică profesională în domeniu, precum și de securitate și sănătate în muncă.</p> <p>RA1 - Studentul/absolventul arată spirit de inițiativă și acțiune pentru actualizarea cunoștințelor profesionale, economice și de cultură organizațională.</p> <p>RA3 - Studentul/absolventul este angajat în învățarea pe tot parcursul vieții pentru dobândirea și implementarea cunoștințelor, după cum este necesar, folosind strategii de învățare adecvate.</p>

* C – cunoștințe; A – aptitudini; RA – responsabilitate și autonomie.

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

6.1. Obiectivul general al disciplinei	➤ Prezentarea principalelor paradigme de programare aplicate la dezvoltarea produselor software și deprinderea de abilități concrete de utilizare a lor.
6.2. Obiectivele specifice	La sfârșitul cursului, studentul va fi capabil

	<ul style="list-style-type: none"> ➤ să definească conceptele fundamentale ale paradigmei programării neprocedurale, respectiv procedurale; ➤ să explice conceptele fundamentale ale diferitelor paradigme de programare; ➤ să aplice conceptele fundamentale ale diferitelor paradigme de programare în anumite limbaje de programare.
--	--

7. Conținuturi

7.1. Curs	Nr. ore	Metode de predare	Observații
Elemente introductive. Programarea structurată	2	interactivă și convențională, centrată pe student	Suport de curs în format electronic (slide-uri)
Paradigma programării procedurale	2	interactivă și convențională, centrată pe student	Suport de curs în format electronic (slide-uri)
Paradigma programării orientate pe obiecte	4	interactivă și convențională, centrată pe student	Suport de curs în format electronic (slide-uri)
Paradigma programării neprocedurale. Paradigma programării declarative	2	interactivă și convențională, centrată pe student	Suport de curs în format electronic (slide-uri)
Paradigma programării logice	4	interactivă și convențională, centrată pe student	Suport de curs în format electronic (slide-uri)
Paradigma programării funcționale	4	interactivă și convențională, centrată pe student	Suport de curs în format electronic (slide-uri)
Paradigma programării orientate pe evenimente	4	interactivă și convențională, centrată pe student	Suport de curs în format electronic (slide-uri)
Paradigma programării orientate pe componente	4	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	Suport de curs în format electronic (slide-uri)
Programarea vizuală	2	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	Suport de curs în format electronic (slide-uri)
Bibliografie [1] M. Oprea (2020), <i>Programare logică și Programare funcțională – Teorie și Aplicații</i> , Editura Matrix Rom, București. [2] M. Bramer (2013), <i>Logic programming with Prolog</i> , Springer.			

[3] R. Bird (1998), <i>Introduction to Functional Programming</i> , 2nd edition, Prentice Hall, New York. [4] C. Giumale (1997), <i>Programare funcțională - o perspectivă pragmatică</i> , Editura Tehnică, București. [5] N. Constantinescu (2018), <i>Bazele programării procedurale</i> , Editura Universitaria, București. [6] M. Gabbriello, S. Martini (2010), <i>Programming languages: principles and paradigms</i> , Springer. [7] M. Oprea (2017), <i>Programare orientată pe obiecte</i> , Editura Universității Petrol-Gaze din Ploiești. [8] P. Deitel, H. Deitel (2011), <i>Java How to Program</i> , Pearson.			
7.2. Seminar / laborator	Nr. ore	Metode de predare	Observații
Recapitulare programare procedurală în limbajele C/C++	1	clasică, centrată pe student și pe rezultatele învățării, dezbateri studii de caz	îndrumar de laborator
Programarea orientată pe obiecte în limbajul Java	4	clasică, centrată pe student și pe rezultatele învățării	îndrumar de laborator
Programarea logică în limbajul Prolog	3	clasică, centrată pe student și pe rezultatele învățării	îndrumar de laborator
Programarea funcțională în limbajul Haskell	2	dezbateri studii de caz, centrare pe student și pe rezultatele învățării	îndrumar de laborator
Programarea orientată pe evenimente. Programarea orientată pe componente. Aplicații în limbajul Java.	4	dezbateri studii de caz, centrare pe student în relație cu dezvoltarea abilităților practice	îndrumar de laborator
Bibliografie			
[1] M. Oprea (2020), <i>Programare logică și Programare funcțională – Teorie și Aplicații</i> , Editura Matrix Rom, București. [2] M. Oprea (2009), <i>Inteligență artificială - îndrumar de laborator</i> , Editura UPG Ploiești. [3] D. Medak, G. Navratil (2003), <i>Haskell Tutorial</i> , Technical University, Vienna. [4] P. Deitel, H. Deitel (2011), <i>Java How to Program</i> , Pearson.			
7.3. Proiect	Nr. ore	Metode de predare	Observații
Bibliografie			

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

- Conținuturile disciplinei sunt specifice domeniului dezvoltării software-ului și sunt coroborate cu așteptările comunității epistemice, a asociațiilor profesionale și angajatorilor ce activează în acest domeniu.

9. Evaluare

Tip activitate	9.1. Criterii de evaluare	9.2. Metode de evaluare	9.3. Pondere din nota finală
9.4. Curs	Verificare finală	Lucrare scrisă cu subiecte teoretice și aplicații	50%
	Teste la curs	Lucrare scrisă, examinare orală	10%
9.5. Seminar/laborator	Activitate laborator și verificări periodice	Examinare orală și program la calculator	30%

	Verificări pe parcurs	Examinare orală, lucrare scrisă și program la calculator	10%
9.6. Proiect			
9.7. Standard minim de performanță			
➤ Cunoașterea conceptelor fundamentale ale diferitelor paradigme de programare, la nivel teoretic și practic.			

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

17.09.2025 _____ _____ _____

Data avizării în departament

Director de departament
Conf. dr. ing. Pricop Emil

Decan
Conf. dr. ing. Bădicioiu Marius

26.09.2025
