

FIȘA DISCIPLINEI ¹⁾

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	Inginerie Software
2.2. Titularul activităților de curs	Conf. dr. ing. Monica Vlădoiu
2.3. Titularul activităților seminar/laborator	Lector dr. inf. Elia Dragomir
2.4. Titularul activității proiect	-
2.5. Anul de studiu	III
2.6. Semestrul *	5
2.7. Tipul de evaluare	C
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 sinteza.

*** 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							28
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren							20
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri							21
Tutoriat							-
Examinări							-
Alte activități							-
3.10 Total ore studiu individual	69						
3.11. Total ore pe semestru	125						
3.12. Numărul de credite	5						

4. Precondiții (acolo unde este cazul)

4.1. de curriculum	<ul style="list-style-type: none"> ➤ Fundamentele Programării, Algoritmi Fundamentali, Programare Orientată pe Obiecte ➤ Structuri de Date, Metode Avansate de Programare
4.2. de competențe	<ul style="list-style-type: none"> ➤ Cunoașterea programării procedurale (paradigmă, limbaj, fundamente, elemente avansate) ➤ Cunoașterea fundamentelor analizei, proiectării și programării orientate pe obiecte ➤ Cunoașterea principalelor structuri de date si a algoritmilor de prelucrare a acestora

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	➤ sală de curs multimedia cu videoproiector și conexiune la Internet
5.2. de desfășurare a seminarului/laboratorului	➤ sală de laborator informatică care permite dezvoltare de aplicații în Java

6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none">➤ C2.1 Identificarea de metodologii adecvate de dezvoltare a sistemelor software;➤ C2.2 Identificarea și explicarea mecanismelor adecvate de specificare a sistemelor software;➤ C2.3 Utilizarea metodologiilor, mecanismelor de specificare și a mediilor de dezvoltare pentru realizarea aplicațiilor informatice;➤ C2.4 Utilizarea de criterii și metode adecvate pentru evaluarea aplicațiilor informatice;➤ C2.5 Realizarea unor proiecte informatice dedicate;➤ C3.2 Identificarea și explicarea modelelor informatice de baza adecvate domeniului de aplicare;➤ C3.3 Utilizarea modelelor și instrumentelor informatice și matematice pentru rezolvarea problemelor specifice domeniului de aplicare;➤ C4.2 Interpretarea de modele matematice și informatice (formale);➤ C4.3 Identificarea modelelor și metodelor adecvate pentru rezolvarea unor probleme reale;➤ C4.5 Încorporarea de modele formale în aplicații specifice din diverse domenii.
Competențe transversale	<ul style="list-style-type: none">➤ CT1. Aplicarea regulilor de muncă organizată și eficientă, a unor atitudini responsabile față de domeniul didactic, științific și profesional, în vederea valorificării creative a propriului potențial, cu respectarea principiilor și normelor de etică profesională;➤ CT2. Desfășurarea eficientă a activităților organizate în echipă și dezvoltarea capacităților empatică și de comunicare inter-personală, de relaționare și colaborare cu persoane și grupuri diverse implicate în dezvoltarea și utilizarea de sisteme software;➤ CT3. Utilizarea de metode și tehnici eficiente de învățare, informare, cercetare și dezvoltare a capacităților de valorificare a cunoștințelor, dar și de adaptare la cerințele unei societăți dinamice și în continuă schimbare, precum și dezvoltarea capacității de a comunica eficient și profesionist atât în limba română, cât și într-o limbă de circulație internațională, prin însușirea și folosirea adecvată a limbajului de specialitate.

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

7.1. Obiectivul general al disciplinei	➤ <i>Dobândirea de către studenți a cunoștințelor fundamentale privind dezvoltarea de sisteme mari de programe și aplicarea corespunzătoare a acestora în practica dezvoltării acestor sisteme, alături de deprinderea de a lucra în echipe mici de programatori și de a prezenta convingător rezultatele efortului de echipă.</i>
7.2. Obiectivele specifice	<p><i>După parcurgerea disciplinei studenții vor putea să:</i></p> <ul style="list-style-type: none">➤ Facă diferența între dezvoltarea de programe de mici dimensiuni și abordarea cu metode ingineresti a dezvoltării sistemelor și aplicațiilor software complexe, de mari dimensiuni;➤ Rezume corect principalele caracteristici ale unui pachet software „de încredere”;➤ Descrie și să aplice în mod adecvat principiile Codului ACM de etică și conduită profesională;➤ Descrie și să analizeze comparativ principalele modele ale procesului de dezvoltare de software;➤ Descrie principalele caracteristici, particularități și probleme din managementul dezvoltării de sisteme mari de programe; Explice importanța factorilor umani în ingineria programării;➤ Explice și să aplice adecvat principiile ingineriei sistemelor și aplicațiilor software complexe, ale specificării, proiectării, dezvoltării și validării lor, în contextul folosirii unor platforme integrate și sub incidența unui management adecvat;➤ Dezvolte pachete software integrate cu structuri de date fundamentale, folosind limbajul Java;➤ Lucreze în cadrul unor echipe mici de programatori, pentru a finaliza și prezenta corespunzător un proiect software atât către management, cât și către client.

8. Conținuturi

8.1. Curs	Nr. ore	Metode de predare	Obs.
1. Ingineria programării (inginerie software - ISW). Software bine făcut. Diversitatea sistemelor software. Activități și principii ale ISW. Încrederea în sistemele software. Etica ISW	6	<ul style="list-style-type: none"> • prelegeri active și angajante; • învățare centrată pe student; • învățare hibridă, cu folosirea resurselor educaționale open; • învățare reflectivă etc. 	
2. Modele ale procesului software. Modelul cascadă. Dezvoltarea incrementală și iterativă. ISW orientată pe reutilizare. Prototipizarea. Modelul spirală al lui Boehm. Procesul unificat Rational. Metode și procese de tip Agile (eXtreme Programming, Scrum). Free/Open Source Software.	4		
3. Managementul proiectelor software. Generalități, particularități și obiective. Funcții și activități. Factori umani. Structuri manageriale. Productivitatea programatorului. Planificarea și realizarea orarului. Mentenanța și evoluția software-ului. Managementul configurațiilor. Documentația. Asigurarea calității.	4		
4. Proiectarea sistemelor de programe. Modelarea proiectării sistemelor software - proiectare orientată pe obiecte, proiectare orientată pe funcții. Proiectarea interfeței utilizator.	4		
5. Specificația sistemelor de programe - Fundamente ale specificării programelor. Specificarea cerințelor. Validare și prototipizare. Specificarea software-ului (axiomatică, algebrică, bazată pe model)	2		
6. Validarea sistemelor de programare - Verificarea și validarea programelor. Tehnici de testare. Verificare statică. Utilitare pentru testare și depanare	2		
7. Practici de programare, tehnici și medii integrate de dezvoltare.	2		
Bibliografie <ol style="list-style-type: none"> 1. Braude, E. J., <i>Software engineering: Modern approaches</i>, New Jersey, John Wiley & Sons, 2011* 2. Ghezzi, C., <i>Fundamentals of Software Engineering</i>, New Jersey, Prentice Hall, 2003* 3. Jones, C., <i>Software engineering best practices: Lessons from successful projects in top companies</i>, McGraw Hill, 2010* 4. McConnell, S., <i>Code Complete: A practical handbook of software construction</i>, Microsoft Press, 2004* 5. Sommerville, I., <i>Software Engineering</i>, Pearson, Boston New York, 2011* 6. Pillone, D., <i>Head first software development</i>, O'Reilly 2008* 7. Vlădoiu, M., Constantinescu Z., Moise, G., <i>Ingineria Programării. Fundamente</i>, Ed. UPG Ploiești, 2015* 8. Resurse educaționale disponibile la http://www.unde.ro/cursuri/IP/ și http://www.unde.ro/cursuri/OCW/ <p>* Disponibile la biblioteca departamentului iTIMF</p>			
8.2. Seminar / laborator	Nr. ore	Metode de predare	Obs.
1. Programare orientată pe obiecte – noțiuni fundamentale, clase și obiecte, attribute și metode, moștenire și polimorfism. Un exemplu de complexitate medie în C++ și Java. Comparatie. Implementare folosind NetBeans. Operații de intrare-ieșire.	2	<ul style="list-style-type: none"> • prezentări active și angajante; • învățarea prin descoperire; • învățare bazată pe rezolvarea de probleme; • învățare centrată pe student; • learning by doing; • învățare hibridă, cu folosirea resurselor educaționale deschise; • învățare reflectivă etc. 	
2. Programare în Java în linie de comandă. Operații I/O.	2		
3, 4. Tablouri în Java. Implementare în NetBeans și în linie de comandă.	4		
5. Sortări în Java (metoda bulelor, prin inserție, prin selecție).	2		
6, 7. Liste înlănțuite în Java. Lista cu header (cu două capete). Iteratori în Java. Sortarea listelor prin inserție.	4		
8, 9. Stive, cozi în Java. Implementare cu tablouri/liste înlănțuite.	4		
10. Liste dublu înlănțuite în Java. Alegere teme proiect.	2		
11. Arbori binari în Java - implementare cu referințe. Construcție și parcurgere RSD (preordine). Calculul înălțimii. Căutarea unei chei.	2		

Alegere teme proiect.			
12. Arbori binari în Java Construcție și parcurgere SRD (inordine) și SDR (postordine). Verificarea echilibrării arborelui.	2		
13, 14. Dezvoltarea unei interfețe grafice utilizator, folosind elemente de design (etichete, butoane, casete de validare, liste de opțiuni, zone de text, layout managers etc.). Integrarea unui set de programe, pentru a crea o aplicație Java unică.	4		
Bibliografie			
1. Waite M., Lafore R., <i>Structuri de date și algoritmi în Java</i> , Teora, 2000*			
2. Lemay L., Cadenhead R., <i>Java 2 fără profesor în 21 de zile</i> , Teora, 2000 **			
3. Chan M. C., Griffith S. W., Iasi A. F., <i>Java - 1001 secrete pentru programatori</i> , Teora, 2000 **			
4. Martin, R.C., <i>Clean code: A handbook of agile software craftsmanship</i> , Prentice Hall, 2009*			
5. Sedgewick, R., <i>Introduction to programming in Java. An interdisciplinary approach</i> , Pearson Education, 2014*			
6. Resurse educationale disponibile la http://www.unde.ro/cursuri/IP/ și http://www.unde.ro/cursuri/OCW/			
* Disponibile la biblioteca departamentului iTIMF ** Disponibile la CERTIMF (L4)			
8.3. Proiect	Nr. ore	Metode de predare	Obse rvații
-			

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

<ul style="list-style-type: none"> • Conținuturile disciplinei corespund cu așteptările reprezentanților comunității epistemice, asociațiilor profesionale și angajatorilor reprezentativi din domeniul aferent programului, așa după cum rezultă din prezenta fișă, dar și din fișa specializării, acestea fiind în concordanță deplină cu CNCIS și COR; • Disciplina de față respectă recomandările IEEE/CS și ACM legate de planul de învățământ și de conținuturile necesare pentru specializarea Informatică/Știința Calculatoarelor; • Discipline având conținuturi educaționale asemănătoare există în planurile de învățământ pentru specializarea Informatică ale marilor universități din România și din străinătate.
--

10. Evaluare

Tip activitate	10.1. Criterii de evaluare	10.2. Metode de evaluare	10.3. Pondere din nota finală
10.4. Curs	Dobândirea competențelor profesionale și transversale specifice disciplinei	<i>Proiect</i> : dezvoltare sistem software în echipa mică de programatori, aplicație informatică integrată și documentație <i>Examinare orală</i> pe teme fundamentale din conținutul cursului	Documentație 45% Aplicație informatică 30% Examinare orală 15% Din oficiu 10%
10.5. Seminar /laborator			
10.6. Proiect			
10.7. Standard minim de performanță			
➤ <i>Realizarea și prezentarea proiectului dezvoltat care să conțină minim aplicația software, declarație de intenții, orar, motivație alegere tip echipă, alocare resurse și documentație utilizator.</i>			

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

09.09.22

Data avizării în departament

21.09.22

Director de departament
(funcție didactică, nume, prenume)
(Semnătură)

Decan
(funcție didactică, nume, prenume)
(Semnătură)