

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	Structuri de Date
2.2. Titularul activităților de curs	Conf. dr. ing. Monica Vlodoiu
2.3. Titularul activităților seminar/laborator	Lector dr. Daniela Schiopu
2.4. Titularul activității proiect	-
2.5. Anul de studiu	II
2.6. Semestrul *	3
2.7. Tipul de evaluare	E
2.8. Categoria formativă** / regimul*** disciplinei	DF/ 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	5	din care: 3.2. curs	2	3.3. Seminar/laborator	3	3.4. Proiect	-
3.5. Total ore din planul de învățământ	70	din care: 3.6. curs	28	3.7. Seminar/laborator	42	3.8. Proiect	-
3.9. Distribuția fondului de timp							ore
Studiu după manual, suport de curs, bibliografie și notițe							20
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren							15
Pregătire seminarii/laboratoare, teme, referate, portofolii și eseuri							20
Tutoriat							-
Examinări							-
Alte activități							-
3.10 Total ore studiu individual	55						
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, Metode Avansate de Programare ➤ Algoritmi Fundamentali
4.2. de competențe	<ul style="list-style-type: none"> ➤ Cunoașterea programării procedurale (paradigmă, limbaj, fundamente, elemente avansate) ➤ Cunoașterea noțiunilor fundamentale privind complexitatea algoritmilor

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 programe în limbajul C

6. Competențe specifice acumulate

Competențe profesionale	<ul style="list-style-type: none">➤ C1.3 Elaborarea codurilor sursă adecvate și testarea unitară a unor componente într-un limbaj de programare cunoscut, pe baza unor specificații de proiectare date.➤ C1.5 Dezvoltarea de unități de program și elaborarea documentațiilor aferente.➤ C2.1 Identificarea de metodologii adecvate de dezvoltare a sistemelor software➤ C2.3 Utilizarea metodologiilor, mecanismelor de specificare și a mediilor de dezvoltare pentru realizarea aplicațiilor informatice➤ C3.1 Descrierea de concepte, teorii și modele folosite în domeniul de aplicare.➤ 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 despre structurile de date elementare și algoritmi folosiți în dezvoltarea de software de bună calitate, cunoștințe referitoare la înțelegerea nevoii de structuri de date, la abstractizare și tipuri de date abstracte, precum și la implementarea acestora ca structuri de date (liste, stive, cozi, arbori, grafuri etc.) și la alegerea și folosirea lor, alături de algoritmi cei mai potriviți, pentru dezvoltarea de aplicații informatice complexe care funcționează eficient.</i>
7.2. Obiectivele specifice	<p><i>După parcurgerea disciplinei studenții vor putea să:</i></p> <ul style="list-style-type: none">➤ Explice de ce sînt necesare tipurile de date abstracte, respectiv structurile de date, și care este legătura dintre ele; descrie parțial corect și să implementeze corespunzător (în limbajul C) tipurile de date abstracte fundamentale (liste, stive, cozi, arbori, grafuri), dar și să descrie tipuri noi pe baza acestora;➤ Compare principalele implementări ale tipurilor de date abstracte fundamentale ca structuri de date, după complexitatea operațiilor elementare asupra acestora, și, în consecință, să aleagă cele mai potrivite structuri de date și cei mai potriviți algoritmi pentru a dezvolta eficient un anumit program sau pachet de programe;➤ Explice importanța alegerii celor mai potrivite structuri de date, respectiv algoritmi, pentru rezolvarea unei probleme de date, care poate fi „soluționată” cu ajutorul unui program de

	calculator, dar și să evalueze critic diversele posibilități și să o aleagă pe cea mai bună într-un anumit context.
--	---

8. Conținuturi

8.1. Curs	Nr. ore	Metode de predare	Obs.
1. Introducere - De ce avem nevoie de structuri de date. Tipuri de date abstracte elementare. Etapele elaborării unui program pentru rezolvarea unei probleme „computerizabile”.	2	<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. Tipuri de date abstracte (TDA) - Conceptul de TDA. Descrierea axiomatică a TDA. Metodă de construcție a unei prezentări de TDA parțial corect. Definiții axiomatice parțial corecte pentru TDA elementare	4		
3. Liste - Concepte de bază. TDA listă. Implementări posibile (vector, înlănțuite: simplă, cu header). Complexități. Funcții elementare de lucru cu liste. Liste sortate. Liste dublu înlănțuite. Liste de liste. Aplicație complexă cu liste circulare	4		
4. Stive și cozi - Concepte de bază pentru stive. TDA stivă. Implementări posibile (vector, înlănțuită). Complexități. Funcții elementare de lucru cu stive. Aplicație complexă cu stive: evaluarea expresiilor aritmetice Concepte de bază pentru cozi. TDA coadă. Implementări posibile (vector circular, listă cu header). Complexități. Funcții elementare de lucru cu cozi. Aplicație complexă cu cozi: sortarea după ranguri	6		
5. Arbori - Concepte de bază. TDA arbore. Implementări posibile (vector tată, cu pointer-i). Complexități. Funcții elementare de lucru cu arbori. Aplicație complexă cu arbori multicăi	6		
6. Grafuri - Concepte de bază. TDA graf. Implementări posibile (matrice sau liste de adiacență). Complexități. Funcții elementare de lucru cu grafuri. Aplicații	2		
7. Alte tipuri de date abstracte. Tabele de dispersie. Dicționare. Heap-uri. Arbori: echilibrați, AVL, multicăi, bicolori, TRIE.	2		
8. Alegerea celei mai potrivite structuri de date și a celui mai potrivit algoritm. Complexități. Comparație.	2		
Bibliografie <ol style="list-style-type: none"> 1. Cormen, T. H. et al., <i>Introduction to Algorithms</i>, MIT Press, 2009* 2. Cristea V., Athanasiu I., Kalisz E., Iorga V., <i>Tehnici de programare</i>, Teora, 1993 (biblioteca UPG) 3. Knuth D.E., <i>The Art of Computer Programming Vol3: Sorting and searching</i>, Addison Wesley, Reading, Mass., 1968, Ed. Tehnică, 1976 (biblioteca UPG, în limba română) 4. Knuth D.E., <i>The Art of Computer Programming. Vol1: Fundamental Algorithms</i>, Addison Wesley, Reading, Mass., 1968, Ed. Tehnică, 1976 (biblioteca UPG, în limba română) 5. Livovschi L. Et al., <i>Bazele informaticii. Algoritmi. Elaborare și complexitate</i>, Univ. din București, 1985 (bibl. UPG) 6. Skiena, S. S., <i>The algorithm design manual</i>, Springer-Verlag, 2010* 7. Vlădoiu, M., Constantinescu, Z., Moise, G. <i>Structuri de Date Fundamentale</i>, 2016* 8. Wirth, N., <i>Algorithms + Data Structures = Programs</i>, Prentice Hall, 1976* 9. Wirth, N., <i>Algorithms and Data Structures</i>, Prentice Hall, 1985* 10. Sedgewick, R., Wayne, K., <i>Algorithms</i>. Addison Wesley, 2011* 11. Resurse educaționale disponibile la http://www.unde.ro/cursuri/ASD/ 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. Implementarea TDA elementare (vector, mulțime ordonată)	2	prezentări active și angajante; învățarea prin descoperire; învățare bazată pe rezolvarea de probleme; învățare centrată	
2. Liste. Liste sortate. Liste dublu înlănțuite. Liste de liste.	8		
3. Stive și cozi. Reprezentare, parcurgere și alte prelucrări elementare.	8		

4. Arbori. Reprezentare, parcurgere și alte prelucrări elementare.	12	pe student; learning by doing; învățare hibridă, cu folosirea resurselor educaționale deschise; învățare reflectivă etc.	
5. Grafuri. Reprezentare, parcurgere și alte prelucrări elementare.	8		
6. Alte tipuri de date abstracte. Reprezentare, parcurgere și alte prelucrări elementare.	4		
Bibliografie			
1. Waite M., Lafore R., <i>Structuri de date și algoritmi în Java</i> , Teora, 2000 (biblioteca UPG)			
2. Resurse educaționale disponibile la http://www.unde.ro/cursuri/ASD/ și http://www.unde.ro/cursuri/OCW/			
8.3. Proiect	Nr. ore	Metode de predare	Observații
-			

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>Evaluare prin probă practică:</i> dezvoltarea de programe C pentru manipularea structurilor de date fundamentale folosind algoritmi specifici <i>Evaluare orală:</i> modificarea programului inițial pentru a corespunde unui set de cerințe ușor diferit, analiza de complexitate, eficientizarea programului inițial prin modificarea structurilor de date și/sau a algoritmilor folosiți etc.	Evaluare program C 45% Evaluare orală 45% Din oficiu 10%
10.5. Seminar /laborator			
10.6. Proiect	-		
10.7. Standard minim de performanță			
➤ <i>Dezvoltarea de programe C cu un grad redus de dificultate pentru manipularea unor structuri de date fundamentale (liste, stive, cozi), implementate dinamic, pe baza algoritmilor specifici.</i>			

Data completării

Semnătura titularului de curs

Semnătura titularului de seminar/laborator

Semnătura titularului de proiect

22.09.24

Data avizării în departament

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

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

24.09.24