

FIȘA DISCIPLINEI¹

1. Date despre program

1.1 Instituția de învățământ superior	Universitatea Politehnică Timișoara
1.2 Facultatea ² / Departamentul ³	Facultatea de Inginerie Hunedoara / Inginerie Electrică și Informatică Industrială
1.3 Catedra	—
1.4 Domeniul de studii (denumire/cod ⁴)	ȘTIINȚE INGINEREȘTI APLICATE / 270
1.5 Ciclul de studii	Licenta
1.6 Programul de studii (denumire/cod/calificarea)	INFORMATICĂ INDUSTRIALĂ / 50 / Inginer

2. Date despre disciplină

2.1 Denumirea disciplinei/Categoria formativă ⁵	Proiectarea algoritmilor / DD						
2.2 Titularul activităților de curs	Conf.Dr. Muscalagiu Ionel						
2.3 Titularul activităților aplicative ⁶	Conf.Dr. Muscalagiu Ionel						
2.4 Anul de studii ⁷	II	2.5 Semestrul	I	2.6 Tipul de evaluare	E	2.7 Regimul disciplinei ⁸	DI

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate)⁹

3.1 Număr de ore asistate integral/săptămână	5 , format din:	3.2 ore curs	2	3.3 ore seminar/laborator/proiect	3
3.1* Număr total de ore asistate integral/sem.	70 , format din:	3.2* ore curs	28	3.3* ore seminar/laborator/proiect	42
3.4 Număr de ore asistate parțial/săptămână	, format din:	3.5 ore practică		3.6 ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/semestru	, format din:	3.5* ore practică		3.6* ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/săptămână	2,14 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			0.14
3.7* Număr total de ore activități neasistate/semestru	30 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			14
		ore studiu individual după manual, suport de curs, bibliografie și notițe			14
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			2
3.8 Total ore/săptămână ¹⁰	7,14				
3.8* Total ore/semestru	100				
3.9 Număr de credite	4				

4. Precondiții (acolo unde este cazul)

4.1 de curriculum	• Programarea calculatoarelor 1, Programarea calculatoarelor 2
4.2 de competențe	•

¹ Formularul corespunde Fișei Disciplinei promovată prin OMECTS 5703/18.12.2011 și cerințelor Standardelor specifice ARACIS valabile începând cu 01.10.2017.

² Se înscrie numele facultății care gestionează programul de studiu căruia îi aparține disciplina.

³ Se înscrie numele departamentului căruia i-a fost încredințată susținerea disciplinei și de care aparține titularul cursului.

⁴ Se înscrie codul prevăzut în HG nr.140/16.03.2017 sau în HG similare actualizate anual.

⁵ Disciplina se încadrează potrivit planului de învățământ în una dintre următoarele categorii formative: disciplină fundamentală (DF), disciplină de domeniu (DD), disciplină de specialitate (DS) sau disciplină complementară (DC).

⁶ Prin activități aplicative se înțeleg activitățile de: seminar (S) / laborator (L) / proiect (P) / practică (Pr).

⁷ Anul de studii în care este prevăzută disciplina în planul de învățământ.

⁸ Disciplina poate avea unul din următoarele regimuri: disciplină impusă (DI), disciplină opțională (DO) sau disciplină facultativă (Df).

⁹ Numărul de ore de la rubricile 3.1*, 3.2*,...,3.8* se obțin prin înmulțirea cu 14 (săptămâni) a numărului de ore din rubricile 3.1, 3.2,..., 3.8. Informațiile din rubricile 3.1, 3.4 și 3.7 sunt chei de verificare folosite de ARACIS sub forma: (3.1)+(3.4) ≥ 28 ore/săpt. și (3.8) ≤ 40 ore/săpt.

¹⁰ Numărul total de ore / săptămână se obține prin însumarea numărului de ore de la punctele 3.1, 3.4 și 3.7.

5. Condiții (acolo unde este cazul)

5.1 de desfășurare a cursului	- Sală de curs echipată cu videoproiector și conexiune la Internet. - Studentii nu se vor prezenta la prelegeri cu telefoanele mobile deschise.
5.2 de desfășurare a activităților practice	- Sală de laborator echipată cu videoproiector și computere. - Studentii nu se vor prezenta la activitățile practice cu telefoanele mobile deschise.

6. Competențe la formarea cărora contribuie disciplina

Competențe specifice	C 1. C 1.1. Identificarea conceptelor de bază proprii științelor ingineresti aplicate; C 1.2. Explicarea structurii și funcționării componentelor diferitelor tipuri de echipamente utilizând teorii și instrumente specifice (scheme, modele matematice, fizice, chimice, biologice etc.); C 1.3. Aplicarea tehnicilor de proiectare și a principiilor de construcție a componentelor diferitelor tipuri de echipamente specifice domeniului și specializării; C 1.4. Utilizarea metodelor de validare a soluțiilor constructive pentru componentele și structurile proiectate; C 1.5. Implementarea de aplicații în practica inginerescă din domeniul specializării, folosind fundamente teoretice ale științelor ingineresti aplicate. C 2. C 2.1. Descrierea structurii și a modului de funcționare a sistemelor informatice în general; C 2.2. Explicarea rolului, funcționalității și utilității sistemelor informatice în general și a sistemelor de prelucrare și gestiune a datelor în domeniul specializării; C 2.3. Utilizarea componentelor software ale sistemelor informatice, folosind algoritmi, protocoale, limbaje, structuri de date; C 2.4. Aprecierea caracteristicilor și calității sistemelor informatice; C 2.5. Prelucrarea și gestionarea datelor utilizând sisteme informatice dedicate. •
Competențele profesionale în care se înscriu competențele specifice	- C 1. Utilizarea adecvată a fundamentelor teoretice ale științelor ingineresti aplicate. - C 2. Utilizarea sistemelor informatice de prelucrare și gestiune a datelor. •
Competențele transversale în care se înscriu competențele specifice	•

7. Obiectivele disciplinei (asociate competențelor de la punctul 6)

7.1 Obiectivul general al disciplinei	Scopul acestei discipline este de a prezenta conceptele programării structurate, structurile de date cu principalele tehnici de programare.
7.2 Obiectivele specifice	Obiectivele specifice ale acestei discipline sunt: - studierea conceptului de tip abstract de date și a celor mai frecvent utilizate tipuri abstracte de date folosite în dezvoltarea aplicațiilor; - studierea structurilor de date cu care se pot implementa aceste tipuri abstracte de date: tablouri, liste înălțuite, grafuri, arbori binari, etc.; - formarea deprinderilor de a proiecta și realiza aplicații pornind de la utilizarea tipurilor abstracte de date; - formarea deprinderilor de a prelucra date stocate în diverse structuri de date: tablouri, articole, liste înălțuite, stive, cozi, arbori și grafuri; - formarea abilităților în proiectarea și implementarea algoritmilor care prelucrează aceste structuri de date;

	• prezentarea unor metode speciale de programare: backtracking, divide et impera, greedy, branch and bound și programare dinamică, cu aprecierea eficienței și performanțelor acestora în raport cu viteza de execuție și spațiul de memorie necesar; • prezentarea principalelor tehnici de sortare și căutare.
--	---

8. Conținuturi¹¹

8.1 Curs	Număr de ore	Metode de predare ¹²
1. Colecții de date 1.1. Liste liniare simplu înlănțuite 1.1.1. Liste liniare simplu înlănțuite ordonate 1.1.2. Stive 1.1.3. Cozi 1.2. Liste liniare dublu înlănțuite 1.2.1. Coadă cu două capete 1.3. Coadă cu priorități 1.4. Liste circulare 1.5. Tabele de dispersie	6	Studentii au acces la curs în format electronic. Se vor utiliza atât prezentări interactive cât și tradiționale. Se vor folosi: problematizarea, studiu de caz, conversația.
2. Grafuri 2.1. Grafuri neorientate 2.1.1. Reprezentarea grafurilor neorientate 2.1.2. Graf parțial, subgraf 2.1.3. Parcurgerea grafurilor neorientate. Conexitate 2.1.4. Clase speciale de grafuri: grafuri complete, grafuri bipartite, grafuri tripartite, grafuri multipartite, graf stea, etc. 2.2. Grafuri orientate 2.2.1. Reprezentarea grafurilor orientate 2.2.2. Parcurgerea grafurilor orientate. Conexitate și tare conexitate 2.3. Drumuri în grafuri. Algoritmi de determinare a drumurilor optime 2.4. Grafuri euleriene și hamiltoniene. Algoritmi de determinare a drumurilor hamiltoniene 2.5. Cuplaje în grafuri bipartite 2.6. Arbori și păduri. Arbore parțial de cost minim	8	
3. Structuri arborescente 3.1. Reprezentarea și parcurgerea structurilor arborescente 3.2. Arbori binari 3.2.1. Arbori binari de căutare 3.2.2. Arbori binari de căutare echilibrați 3.3. Arbori ternari	4	
4. Metode speciale de programare 4.1. Metoda "divide et impera" 4.2. Metoda "greedy" 4.3. Metoda "backtracking" 4.4. Metoda "branch and bound" 4.5. Metoda programării dinamice 4.6. Tehnici de sortare și căutare	8	
5. Complexitatea algoritmilor	2	

¹¹ Se detaliază toate activitățile didactice prevăzute prin planul de învățământ (tematicile prelegerilor și ale seminariilor, lista lucrărilor de laborator, conținuturile etapelor de elaborare a proiectelor, tematica fiecărui stagi de practică). Titlurile lucrărilor de laborator care se efectuează pe standuri vor fi însoțite de notația „(*)”.

¹² Prezentarea metodelor de predare va include și folosirea noilor tehnologii (e-mail, pagină personalizată de web, resurse în format electronic etc.).

Bibliografie¹³

1. Anca-Elena Iordan, Manuela Pănoiu, Structuri de date și algoritmi, Editura Politehnica, Timișoara, 2010
2. Cristian Giumale, Introducere în analiza algoritmilor, Editura Polirom, 2004, Iași
3. Dorel Lucanu, Mitică Craus, Proiectarea algoritmilor, Editura Polirom, Iași, 2008
4. Iosif Ignat, Claudia Lavinia Ignat, Structuri de date și algoritmi, Editura Albastră, Cluj-Napoca, 2007
5. Vladimir Crețu, Structuri de date și algoritmi, Editura Politehnica, Timisoara, 2004
6. Valeriu Iorga, Colecții de date, Editura Albastră, Cluj-Napoca, 2012
7. Muscalagiu I, Curs în format electronic, Campus virtual UPT, <https://cv.upt.ro/course/view.php?id=3566>

8.2 Activități aplicative ¹⁴	Număr de ore	Metode de predare
Seminar	14]	Se va utiliza exercițiul la tablă și implementarea programului utilizând calculatorul.
1. Implementarea în C++ a tipului generic listă simplu înlănțuită	[2	
2. Implementarea în C++ a tipului generic listă dublu înlănțuită	2	
3. Implementarea în C++ a tipurilor generice coadă și tabel de dispersie	2	
4. Implementarea în C++ a tipului generic graf neorientat	2	
5. Implementarea în C++ a tipului generic graf orientat	2	
6. Implementarea în C++ a tipului generic arbore binar	2	
7. Implementarea în C++ a metodelor speciale de programare	2	

Laborator	28	
1. Analiza, proiectarea și implementarea de aplicații care utilizează liste simplu înlănțuite / dublu înlănțuite	4	
2. Analiza, proiectarea și implementarea de aplicații care utilizează cozii și tabele de dispersie	2	
3. Analiza, proiectarea și implementarea de aplicații care utilizează grafuri neorientate	2	
4. Analiza, proiectarea și implementarea de aplicații care utilizează grafuri orientate	4	
5. Analiza, proiectarea și implementarea de aplicații care utilizează grafuri euleriene și hamiltoniene	4	
6. Analiza, proiectarea și implementarea de aplicații care utilizează grafuri bipartite	2	
7. Analiza, proiectarea și implementarea de aplicații care utilizează arbori binari și ternari	2	
8. Analiza, proiectarea și implementarea de aplicații care utilizează metode speciale de programare: greedy, programare dinamica, backtracking, divide et impera	2	
9. Complexitatea algoritmilor	6	

Bibliografie¹⁵

1. Anca-Elena Iordan, Manuela Pănoiu, Structuri de date și algoritmi, Editura Politehnica, Timișoara, 2010
2. Valeriu Iorga, Cristian Opincaru, Corina Stratan, Alexandru Chiriță, Structuri de date și algoritmi. Aplicații în C++ folosind STL, Editura Polirom, Iași, 2005
3. Teodor Toadere, Grafe: teorie, algoritmi și aplicații, Editura Albastră, Cluj-Napoca, 2009
4. Vladimir-Ioan Crețu, Structuri de date și algoritmi, Editura "Orizonturi Universitare Mirton, Timișoara, 2011
5. Dumitru Fanache, Teoria algoritmică a grafurilor, Editura Paralela 45, București, 2016
6. Muscalagiu I, Curs în format electronic, Campus virtual UPT, <https://cv.upt.ro/course/view.php?id=3566>

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

¹³ Cel puțin un titlu trebuie să aparțină colectivului disciplinei iar cel puțin un titlu trebuie să se refere la o lucrare de referință pentru disciplină, de circulație națională și internațională, existentă în biblioteca UPT.

¹⁴ Tipurile de activități aplicative sunt cele precizate în nota de subsol 5. Dacă disciplina conține mai multe tipuri de activități aplicative atunci ele se trec consecutiv în liniile tabelului de mai jos. Tipul activității se va înscrie într-o linie distinctă sub forma: „Seminar:”, „Laborator:”, „Proiect:” și/sau „Practică:”.

¹⁵ Cel puțin un titlu trebuie să aparțină colectivului disciplinei.

- Disciplina vine în întâmpinarea așteptărilor angajatorilor reprezentativi din domeniul aferent programului prin conținutul orelor de curs, seminar și laborator.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare ¹⁶	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	Cunoștințe teoretice	Oral - subiecte teoretice și aplicații	0,66
10.5 Activități aplicative	S:		
	L: Abilități în analiza, proiectarea și implementarea aplicațiilor de laborator	Oral – aplicații utilizând calculatorul	0,34
	P ¹⁷ :		
	Pr:		
10.6 Standard minim de performanță (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor ¹⁸)			
• Studentul va promova disciplina dacă reușește să implementeze una dintre aplicațiile propuse. • Programele realizate trebuie să fie măcar compilate corect			

Data completării

10.09.2024

Director de departament
(semnătura)

.....

Titular de curs
(semnătura)

.....

Data avizării în Consiliul Facultății¹⁹

17.09.2024

Titular activități aplicative
(semnătura)

.....

Decan
(semnătura)

.....

¹⁶ Fișele disciplinelor trebuie să conțină procedura de evaluare a disciplinei cu precizarea criteriilor, a metodelor și a formelor de evaluare, precum și cu precizarea ponderilor atribuite acestora în nota finală. Criteriile de evaluare se formulează în mod distinct pentru fiecare activitate prevăzută în planul de învățământ (curs, seminar, laborator, proiect). Ele se vor referi și la formele de verificare pe parcurs (teme de casă, referate ș.a.)

¹⁷ În cazul când proiectul nu este o disciplină distinctă, în această rubrică se va preciza și modul în care rezultatul evaluării proiectului condiționează admiterea studentului la evaluarea finală din cadrul disciplinei.

¹⁸ Nu se va explica cum se acorda nota de promovare.

¹⁹ Avizarea este precedată de discutarea punctului de vedere al board-ului de care aparține programul de studii cu privire la fișa disciplinei.