

 <small>UNIVERSITATEA TEHNICĂ A MOLDOVEI</small>	CURRICULA UNITĂȚII DE CURS/MODULULUI	COD: F.02.O.008
	STRUCTURI DE DATE ȘI ALGORITMI	DATA: 30,05.2017 PAGINA: 1/14

FACULTATEA CALCULATOARE, INFORMATICA ȘI MICROELECTRONICA

DEPARTAMENTUL INGINERIA SOFTWARE ȘI AUTOMATICĂ

APROBATĂ

la ședința DISA
nr. 11 din 02 iunie 2017
Șeful departamentului ISA
Dumitru CIORBĂ, conf. univ., dr.

APROBATĂ

la ședința Consiliului FCIM
nr. 7 din 28 iunie 2017
Președintele Consiliului FCIM
Ion BALMUȘ, conf. univ., dr.

Program de studiu: 0613.3 Ingineria software

Denumirea unității de curs: STRUCTURI DE DATE ȘI ALGORITMI

Benefeciari: Studenții anului I, învățământ cu frecvență

Ciclul de învățământ: Studii superioare de Licență, ciclul I

Numărul de credite ECTS: 5 (75 ore în auditoriu și 75 ore de activități individuale ale studentului, 1 credit = 15 ore de activități în auditoriu și 15 ore de activități individuale ale studentului)

Titularul unității de curs: conf. unv., dr. Mihail KULEV

semnătura titularului de curs

I.PRELIMINARII

Disciplina Structuri de date și algoritmi este o disciplină, care se bazează pe disciplina fundamentală Programarea calculatoarelor. Disciplina aduce o contribuție importantă la pregătirea

 <small>UNIVERSITATEA TEHNICĂ A MOLDOVEI</small>	CURRICULA UNITĂȚII DE CURS/MODULULUI	COD: F.02.O.008
	STRUCTURI DE DATE ȘI ALGORITMI	DATA: 30,05.2017 PAGINA: 2/14

profesională a viitorilor specialiști și totodată conține mai multe elemente de aplicații în diferite domenii. Ea servește și ca element de legătură cu alte obiecte de specializare opționale.

Scopul principal al cursului constă în familiarizarea studenților cu utilizarea structurilor de date și algoritmilor în rezolvarea problemelor din diferite domenii. Se urmărește explicarea și înțelegerea de către studenți a structurilor de date statice și dinamice, tipurilor de date abstracte: tablouri, liste, stive, cozi, arbori. Studiarea algoritmilor și tehnicilor de programare corespunzătoare acestor structuri de date și aplicarea acestor cunoștințe la formularea corectă a problemelor, elaborarea metodelor, algoritmilor și tehnicilor de rezolvare ale problemelor.

Unitatea de curs „Structuri de date și algoritmi” este inclusă în categoria unităților de curs de formare fundamentală pentru studenții programului de studiu **0613.3 Ingineria software..** Consolidarea materialului teoretic și obținerea abilităților practice se realizează în procesul de realizare a lucrărilor de laborator, efectuării lucrărilor practice și a proiectelor de licență.

Obiectivele principale ale cursului „Structuri de date și algoritmi” reprezintă formarea la studenți a următoarelor abilități:

- ✓ Cunoșterea profundă construcțiilor complicate ale limbajelor C: tipuri de date structurate statice și dinamice, funcții, tehnici de programare etc.
- ✓ Insusirea mecanismelor de stocare, structurare și prelucrare a datelor cu alcatuire complexă.
- ✓ Cunoșterea principiilor de bază utilizate în alcatuirea algoritmilor ca etapă esențială în dezvoltarea eficientă a aplicațiilor software.
- ✓ Cunoșterea tehnicilor de programare și a algoritmilor ce țin de probleme de stocare, sortare, căutare și optimizare a datelor.
- ✓ Abilitatea de a aplica cunoștințele generale privind structurile de date și structurarea algoritmică a programelor
- ✓ Posibilitatea de a evalua eficiența implementării unei aplicații software folosind criteriile de performanță analizate la curs.
- ✓ De utilizare a limbajelor și mediilor de programare mai performante.

Cursul este orientat spre pregătirea specialiștilor de o calificare înaltă în domeniul ingineriei software și automatizării.

II. PRECONDIȚII DE ACCES LA UNITATEA DE CURS/MODUL:


Pentru a atinge obiectivele cursului studenții trebuie să posede bazele informaticii, programării structurate. Aceste competențe sunt formate de următoarele unități de curs, prevăzute de planul de învățământ: Matematica superioară, Matematici speciale, Programarea calculatoarelor.

III. COMPETENȚELE CARE URMEAZĂ A FI DEZVOLTATE

Competențele formate de această unitate de curs vor servi ca bază pentru formarea competențelor profesionale în cadrul unităților de curs: Metode și modele de calcul, Analiza și proiectarea algoritmilor, Programarea orientată pe obiecte, Analiza și modelarea orientată pe obiecte, Programarea în rețea, Elaborarea tezelor de licență.

Unitatea de curs prevede formarea următoarelor competențe profesionale și transversale:

C1. Privind fundamentele științifice și ingineresti ale tehnologiilor informaționale

	CURRICULA UNITĂȚII DE CURS/MODULULUI				COD: F.02.O.008
	STRUCTURI DE DATE ȘI ALGORITMI				DATA: 30,05.2017 PAGINA: 3/14

C1.1 Identificarea și definirea conceptelor, teoriilor și metodelor de științe fundamentale și aplicative suport pentru ingineria tehnologiilor informaționale

C1.2 Explicarea soluțiilor ingineresti prin utilizarea tehnicilor, conceptelor și principiilor din științele exacte și aplicative

C1.3 Rezolvarea problemelor din domenii de activitate umană prin aplicarea în special al tehnicilor și metodelor de calcul numeric

C1.4 Alegerea criteriilor și metodelor pentru analiza avantajelor și dezavantajelor metodelor și procedeele aplicate la soluționarea problemelor de calcul numeric

C1.5 Modelarea unor probleme tip din științele aplicative folosind aparatul matematic

CT1. Comportarea onorabilă, responsabilă, etică, în spiritul legii pentru a asigura îndeplinirea sarcinilor profesionale.

CT2. Demonstrarea capacității de lucru în echipă, identificarea rolurilor și responsabilităților individuale și comune, luarea deciziilor și atribuirea de sarcini, cu aplicarea de tehnici de relaționare și muncă eficientă în cadrul echipei.

CT3. Demonstrarea spiritului de inițiativă și acțiune pentru dezvoltarea profesională și personală, prin formare continuă utilizând surse de documentare în limba română și în limbile de circulație internațională.

IV. ADMINISTRAREA UNITĂȚII DE CURS

Codul disciplinei	Anul predării	Semestrul	Numărul de ore				Evaluarea		
			Prelegeri	Seminare	Lucrări de proiectare (de laborator)	Lucrul individual	Credite	Curentă	Finală
F.02.O.008	Învățământ cu frecvență								
	I	2	30	30	15	75	5	2 atestări	examen

V. REZULTATELE ÎNVĂȚĂRII, CONȚINUTURI ȘI METODE DIDACTICE APLICATE

Rezultatele învățării. Studentul trebuie:	Conținuturi		Metode de predare	Realizarea în timp (ore)*			
	Prelegeri	Lucrări de laborator		Învățământ cu frecvență		Învățământ cu frecvență redusă	
				prelegeri	lab/ prac	prelegeri	lab/ sem
1	2	3	4	5	6	7	8
<p>să cunoască:</p> <ul style="list-style-type: none"> ▪ Obiectivele și scopul disciplinei; ▪ Clasificarea tipurilor de date; ▪ Definiția tipului abstract de date; ▪ Etapele implementării tipului de date abstract în limbajul C. <p>să fie capabil:</p> <ul style="list-style-type: none"> ▪ Să elaboreze algoritmi și coduri de prelucrare tablourilor de structuri. 	<p>Tema 1. INTRODUCERE Introducere. Principalele definiții și concepția structurilor de date (SD). Clasificare și exemple: liste, stive, arbori etc. . Structura de date, tipul de date și tipul de date abstract. Etapele implementării tipului de date abstract în limbajul C.</p>	<p>Lucrarea practică nr.1. Etapele implementării tipului de date abstract în limbajul C. Tipul de date abstract „Tablou de structuri”.</p> <p>Lucrarea de proiectare (de laborator) nr.1. Implementarea tipului de date abstract “Tablou de structuri” in limbajul C.</p>	<p>Pentru prelegere: expunerea interactivă convențională.</p> <p>Pentru lucrare practică: algoritmi, exemple de cod C, prezentare, analiza, discuție</p> <p>Pentru lucrare de proiectare (de laborator): Prezentarea și rularea unui program - exemplu. Asistarea și colaborarea cu studenți în efectuarea lucrării de laborator.</p>	4	2/2		

1	2	3	4	5	6	7	8
<p>să cunoască:</p> <ul style="list-style-type: none"> ▪ Clasificarea listelor și caracteristici generale; ▪ Tipul de date abstract „Lista simplu înlănțuită”; ▪ Crearea și parcurgerea listei simplu înlănțuite; ▪ Operațiile specifice prelucrării listelor. <p>să fie capabil:</p> <ul style="list-style-type: none"> ▪ Să aplice corect operațiile specifice prelucrării listelor; ▪ Să aplice corect funcțiile pentru prelucrarea listelor simple înlănțuite; ▪ Să elaboreze algoritmi și coduri de prelucrare listelor simple înlănțuite. 	<p>Tema 2. LISTE ÎNLĂNȚUITE Clasificarea și caracteristici generale. Lista simplu înlănțuită. Implementarea tipului de date abstract „Lista simplu înlănțuită” în limbajul C. Parcurgerea listei simplu înlănțuite necirculare și circulare folosind algoritmi iterativi. Algoritm iterativ pentru crearea listei simplu înlănțuite cu n elemente. Implementarea funcțiilor pentru prelucrarea listei simplu înlănțuite.</p>	<p>Lucrarea practică nr.2. Tipul de date abstract „Lista simplu înlănțuită”. Implementarea funcțiilor pentru prelucrarea listei simplu înlănțuite.</p> <p>Lucrarea de proiectare (de laborator) nr.2. Implementarea tipului de date abstract “Lista simplu inlanțuită” in limbajul C. Partea I.</p>	<p>Pentru prelegere: expunerea interactivă, prezentări, exemplificarea, învățare prin colaborare.</p> <p>Pentru lucrare practică: algoritmi, exemple de cod C, prezentare, analiza, discuție.</p> <p>Pentru lucrare de proiectare (de laborator): prezentarea și rularea unui program - exemplu. Asistarea și colaborarea cu studenți în efectuarea lucrării de laborator.</p>	4	3/3		

1	2	3	4	5	6	7	8
<p>să cunoască:</p> <ul style="list-style-type: none"> ▪ Definiția și clasificarea structurilor dinamice de date; ▪ Funcții predefinite C pentru alocarea dinamică a memoriei; ▪ Operațiile asupra structurilor de date dinamice liniare; ▪ Tehnici de dezvoltare aplicațiilor cu utilizarea structurilor dinamice de date liniare. <p>să fie capabil:</p> <ul style="list-style-type: none"> ▪ Să elaboreze aplicații cu utilizarea structurilor de date dinamice: liste simplu și dublu înlanțuite, liste cu priorități, stive, cozi. 	<p>TEMA 3. LISTA DUBLU ÎNLĂNȚUITĂ. STIVA. COADA</p> <p>Implementarea tipului abstract de date „Lista dublu înlanțuită” în limbajul C. Algoritm iterativ pentru crearea listei dublu înlanțuite cu n elemente. Implementarea funcțiilor pentru prelucrarea listei dublu înlanțuite. Tipuri abstracte de date stive și cozi.</p>	<p>Lucrarea practică nr.3.</p> <p>Tipul de date abstract „Lista dublu înlanțuită”. Implementarea funcțiilor pentru prelucrarea listei dublu înlanțuite, stivei și cozei.</p> <p>Lucrarea de proiectare (de laborator) nr.3.</p> <p>Implementarea tipului de date abstract “Lista simplu inlanțuită” in limbajul C. Partea II.</p>	<p>Pentru prelegere:</p> <p>expunerea interactivă, prezentări, exemplificarea, învățare prin colaborare.</p> <p>Pentru lucrare practică:</p> <p>probleme, algoritmi, exemple de cod, analiza, dezbateri, lucrul în grup, testarea programului.</p> <p>Pentru lucrare de proiectare (de laborator):</p> <p>Prezentarea și rularea unui program - exemplu. Asistarea și colaborarea cu studenți în efectuarea lucrării de laborator.</p>	6	2/3		

1	2	3	4	5	6	7	8
<p>să cunoască:</p> <ul style="list-style-type: none"> ▪ Definiția și clasificarea arborilor; ▪ Algoritmi de prelucrare a arborilor de diverse tipuri; ▪ Arbore binar de căutare; ▪ Tehnici de dezvoltare aplicațiilor cu utilizarea arborilor de diverse tipuri. <p>să fie capabil:</p> <ul style="list-style-type: none"> ▪ Să elaboreze aplicații cu utilizarea arborilor de diverse tipuri. 	<p>Tema 4. ARBORI MULTICAI. ARBORI BINARI</p> <p>Algoritmi de prelucrare a arborilor de diverse tipuri. Implementarea tipului de date abstract "Arbore binar generalizat" în limbajul C. Algoritmi iterativi și algoritmi recursivi. Arbore binar de căutare. Implementarea funcțiilor pentru prelucrarea arborilor.</p>	<p>Lucrarea practică nr. 4.</p> <p>Tipul de date abstract „Arbore binar generalizat”. Implementarea funcțiilor pentru prelucrarea arborelui binar generalizat. Algoritmi iterativi și recursivi.</p> <p>Lucrarea de proiectare (de laborator) nr.4.</p> <p>Implementarea tipului de date abstract "Arbore binar generalizat" în limbajul C. Algoritmi iterativi.</p> <p>Lucrarea de de proiectare (de laborator) nr. 5.</p> <p>Implementarea tipului de date abstract "Arbore binar generalizat" în limbajul C. Algoritmi recursivi.</p>	<p>Pentru prelegere:</p> <p>expunerea interactivă, prezentări, exemplificarea, învățare prin colaborare.</p> <p>Pentru lucrare practică:</p> <p>probleme, algoritmi, exemple de cod, analiza, dezbateri, lucrul în grup, testarea programului.</p> <p>Pentru lucrare de proiectare (de laborator):</p> <p>Prezentarea și rularea unui program - exemplu. Asistarea și colaborarea cu studenți în efectuarea lucrării de laborator.</p>	6	4/4		

1	2	3	4	5	6	7	8
<p>să cunoască:</p> <ul style="list-style-type: none"> ▪ Criterii de clasificare a metodelor și a algoritmilor de sortare și căutare; ▪ Algoritmi de sortare cu viteza redusă: prin interschimbare,, inserție și selecție; ▪ Algoritmi de căutare secvențială și binară; ▪ Algoritmi de sortare eficientă: Quicksort, Mergesort și Heapsort. <p>să fie capabil:</p> <ul style="list-style-type: none"> ▪ Să elaboreze coduri C pentru implementarea algoritmilor de sortare și căutare. 	<p>Tema 5. ALGORITMI DE CĂUTARE ȘI SORTARE</p> <p>Algoritmi de căutare și sortare. Clasificare și strategii generale de sortare și de căutare. Algoritmi de sortare prin selecție, bulelor, prin inserție și Shell. Algoritmi de sortare cu performanță înaltă – Quicksort, Mergesort și Heapsort. Căutare secvențială și binară. Analiza emperică a algorimilor de sortare și căutare.</p>	<p>Lucrarea practică nr. 5.</p> <p>Algoritmi de căutare și sortare. Analiza emperică a algorimilor de sortare și căutare.</p> <p>Lucrarea de de proiectare (de laborator) nr. 6.</p> <p>Analiza emperică a algoritmilor de sortare și căutare.</p>	<p>Pentru prelegere:</p> <p>expunerea interactivă, prezentări, exemplificarea, învățare prin colaborare.</p> <p>Pentru lucrare practică:</p> <p>probleme, algoritmi, exemple de cod, analiza, dezbateri, lucrul în grup, testarea programului.</p> <p>Pentru lucrare de proiectare (de laborator):</p> <p>Prezentarea și rularea unui program - exemplu. Asistarea și colaborarea cu studenți în efectuarea lucrării de laborator.</p>	4	2/2		

1	2	3	4	5	6	7	8
<p>să cunoască:</p> <ul style="list-style-type: none"> Metoda de căutare optimală Greedy; Algoritmul Backtracking; Tehnica de programare Divide și imperă; Algoritmi euristici, probabalistici, genetici. <p>să fie capabil:</p> <ul style="list-style-type: none"> Să elaboreze aplicații C utilizând algoritmi și tehnicile Divide și Imperă, algoritmul Backtracking pentru dezvoltarea aplicațiilor practice. 	<p>Tema 6. ALGORITMI ȘI METODE DE CĂUTARE OPTIMALĂ A SOLUȚIILOR Clasificarea algoritmilor. Avantaje și dezavantaje ale algoritmilor din punctul de vedere: universalitate, minimum memorie, viteză maximală.</p> <p>Tema 7. TEHNICI ȘI METODE DE PROGRAMARE Algoritmii: Forța brută, Divide et Impera, Programarea Dinamică, Backtracking, Branch and Bound, Greedy, euristici, probabilistici, genetici.</p>	<p>Lucrarea practică nr. 6. Tehnici și metode de programare.</p> <p>Lucrarea de proiectare (de laborator) nr. 7. Tehnici și metode de programare.</p>	<p>Pentru prelegere: expunerea interactivă, prezentări, exemplificarea, învățare prin colaborare.</p> <p>Pentru lucrare practică: probleme, algoritmi, exemple de cod, analiza, dezbateri, lucrul în grup, testarea programului.</p> <p>Pentru lucrare de laborator: Prezentarea și rularea unui program - exemplu. Asistarea și colaborarea cu studenți în efectuarea lucrării de laborator.</p>	<p>2</p> <p>4</p>	<p>-</p> <p>2/1</p>		

 UNIVERSITATEA TEHNICĂ A MOLDOVEI	CURRICULA UNITĂȚII DE CURS/MODULULUI	COD: F.02.O.008
	STRUCTURI DE DATE ȘI ALGORITMI	DATA: 30,05.2017 PAGINA: 10/14

VI. SUGESTII PENTRU ACTIVITATEA INDIVIDUALĂ A STUDENȚILOR

Pe parcursul semestrului, studenții realizează activități individuale, care includ:

- studiul literaturii obligatorii conform listei surselor bibliografice prezentate în curriculum;
- realizarea temelor pentru acasă, propuse în cadrul orelor de seminar și lucrărilor de laborator;

Pe parcursul semestrului studenților li se propune un set de probleme pentru elaborarea aplicațiilor individuale în scopul dezvoltării abilităților de elabora aplicații și proiecte în C.

VII. LISTA DE SUBIECTE PENTRU EVALUĂRI PERIODICE ȘI CEA FINALĂ

CHESTIONAR PENTRU ATESTAREA I

1. Structuri de date și algoritmi. Noțiuni generale și obiectivele cursului
2. Structuri de date. Clasificarea și caracteristici generale
3. Structura de date, tipul de date și tipul de date abstract
4. Etapele implementării tipului de date abstract în limbajul C
5. Tabloul de structuri. Implementarea tipului de date abstract „Tablou de structuri” în limbajul C
6. Liste înlanțuite. Clasificarea și caracteristici generale
7. Lista simplu înlanțuită. Implementarea tipului abstract de date „Lista simplu înlanțuită” în limbajul C
8. Parcurgerea listei simplu înlanțuite necirculare și circulare folosind algoritmi iterativi
9. Algoritm iterativ pentru crearea listei simplu înlanțuite cu n elemente
10. Lista simplu înlanțuită. Algoritmi pentru introducerea și afișarea informației despre elementele listei
11. Lista simplu înlanțuită. Algoritmi pentru căutarea și modificarea elementului
12. Lista simplu înlanțuită. Algoritmi pentru interschimbarea a două elemente și sortarea listei
13. Lista simplu înlanțuită. Algoritmi pentru adăugarea unui element nou la sfârșitul și la începutul listei
14. Lista simplu înlanțuită. Algoritmi pentru înserarea unui element nou în lista
15. Lista simplu înlanțuită. Algoritm pentru eliminarea unui element din lista
16. Lista simplu înlanțuită. Algoritmi pentru divizarea listei și concatenarea a 2 liste
17. Lista simplu înlanțuită. Algoritmi pentru determinarea adresei elementului precedent și a adresei ultimului element al listei
18. Lista simplu înlanțuită. Algoritmi pentru determinarea lungimii listei și eliberarea memoriei
19. Lista dublu înlanțuită. Implementarea tipului abstract de date „Lista dublu înlanțuită” în limbajul C
20. Parcurgerea listei dublu înlanțuite necirculare și circulare folosind algoritmi iterativi
21. Algoritm iterativ pentru crearea listei dublu înlanțuite cu n elemente
22. Lista dublu înlanțuită. Algoritm pentru interschimbarea a două elemente
23. Lista simplu înlanțuită. Algoritmi pentru adăugarea unui element nou la sfârșitul și la începutul listei
24. Lista dublu înlanțuită. Algoritmi pentru înserarea unui element nou în lista
25. Lista dublu înlanțuită. Algoritm pentru eliminarea unui element din lista
26. Lista dublu înlanțuită. Algoritmi pentru divizarea listei și concatenarea a 2 liste

 <small>UNIVERSITATEA TEHNICĂ A MOLDOVEI</small>	CURRICULA UNITĂȚII DE CURS/MODULULUI	COD: F.02.O.008
	STRUCTURI DE DATE ȘI ALGORITMI	DATA: 30,05.2017 PAGINA: 11/14


27. Coada. Implementarea tipului abstract de date "Coadă" în limbajul C
28. Algoritmi pentru înserarea unui element în coada și ștergerea unui element din coada
29. Stiva. Implementarea tipului abstract de date "Stivă" în limbajul C
30. Algoritmi pentru înserarea unui element în stiva și ștergerea unui element din stiva

CHESTIONAR PENTRU ATESTAREA A II-A

1. Arbori. Clasificarea, caracteristica generală. Parcurgerea nodurilor a unui arbore binar
2. Algoritmi iterativi de prelucrare a unui arbore binar folosind parcurgeri pe niveluri (în largime). Descrierea generală
3. Algoritmi recursivi de prelucrare a unui arbore binar folosind parcurgere recursivă. Descrierea generală
4. Algoritm iterativ de prelucrare a unui arbore binar folosind parcurgerea recursivă preordine (în adâncime). Descrierea generală
5. Algoritm iterativ pentru crearea a unui arbore binar folosind parcurgerea pe niveluri (in largime)
6. Arbore binar. Algoritmi pentru introducerea informației despre noduri și căutarea unui nod folosind parcurgere pe niveluri (in largime)
7. Arbore binar. Algoritmi de determinare a numărului de noduri și de eliberare a memoriei folosind parcurgere pe niveluri (in largime)
8. Algoritm recursiv pentru crearea a unui arbore binar folosind parcurgerea recursivă preordine (în adâncime)
9. Arbore binar. Algoritmi recursivi pentru afișarea informații despre noduri
10. Arbore binar. Algoritmi recursivi pentru căutarea unui nod și eliberarea memoriei
11. Arbore binar. Algoritm iterativ pentru determinarea numărului nodurilor arborelui folosind parcurgerea recursivă preordine (în adâncime)
12. Algoritm iterativ pentru crearea a unui arbore binar folosind parcurgerea recursivă preordine (în adâncime)
13. Arbore binar. Algoritmi iterativi pentru afișarea informației despre noduri și căutarea unui nod folosind parcurgerea recursivă preordine (în adâncime)
15. Arbore binar. Algoritm iterativ de eliberare a memoriei folosind parcurgerea recursivă
16. Arbore binar de căutare și implementarea lui în limbajul C
17. Crearea arborelui binar de căutare. Algoritm pentru inserarea unui nod
18. Arbore binar de căutare . Algoritmi pentru căutarea unui nod
19. Algoritmi de căutare secvențială și binară pentru tabloul unidimensional
20. Algoritmi de sortare. Clasificarea și caracteristici generale
21. Algoritm de sortare prin selecție
22. Algoritmi de sortare "bubble" și prin inserție
23. Algoritmi de sortare "shaker" și Shell
24. Algoritm recursiv de sortare prin interclasare "merge"
25. Arbore binar "heap" și coada de prioritate
26. Algoritmi de sortare "coada de prioritate" și "heap"
27. Algoritm recursiv de sortare rapidă quicksort
28. Funcții standard qsort () și bsearch () în limbajul C
29. Tehnici de programare. Clasificare și caracteristici generale
30. Tehnica de programare "Divide et empera". Exemple

CHESTIONAR PENTRU EXAMEN

1. Structuri de date și algoritmi. Noțiuni generale și obiectivele cursului
2. Structuri de date. Clasificarea și caracteristici generale
3. Structura de date, tipul de date și tipul de date abstract
4. Etapele implementării tipului de date abstract în limbajul C
5. Tabloul de structuri. Implementarea tipului de date abstract „Tablou de structuri” în limbajul C
6. Liste înlanțuite. Clasificarea și caracteristici generale
7. Lista simplu înlanțuită. Implementarea tipului abstract de date „Lista simplu înlanțuită” în limbajul C
8. Parcurgerea listei simplu înlanțuite necirculare și circulare folosind algoritmi iterativi
9. Algoritm iterativ pentru crearea listei simplu înlanțuite cu n elemente
10. Lista simplu înlanțuită. Algoritmi pentru introducerea și afișarea informației despre elementele listei
11. Lista simplu înlanțuită. Algoritmi pentru căutarea și modificarea elementului
12. Lista simplu înlanțuită. Algoritmi pentru interschimbarea a două elemente și sortarea listei
13. Lista simplu înlanțuită. Algoritmi pentru adăugarea unui element nou la sfârșitul și la începutul listei
14. Lista simplu înlanțuită. Algoritmi pentru înserarea unui element nou în lista
15. Lista simplu înlanțuită. Algoritm pentru eliminarea unui element din lista
16. Lista simplu înlanțuită. Algoritmi pentru divizarea listei și concatenarea a 2 liste
17. Lista simplu înlanțuită. Algoritmi pentru determinarea adresei elementului precedent și a adresei ultimului element al listei
18. Lista simplu înlanțuită. Algoritmi pentru determinarea lungimii listei și eliberarea memoriei
19. Lista dublu înlanțuită. Implementarea tipului abstract de date „Lista dublu înlanțuită” în limbajul C
20. Parcurgerea listei dublu înlanțuite necirculare și circulare folosind algoritmi iterativi
21. Algoritm iterativ pentru crearea listei dublu înlanțuite cu n elemente
22. Lista dublu înlanțuită. Algoritm pentru interschimbarea a două elemente
23. Lista simplu înlanțuită. Algoritmi pentru adăugarea unui element nou la sfârșitul și la începutul listei
24. Lista dublu înlanțuită. Algoritmi pentru înserarea unui element nou în lista
25. Lista dublu înlanțuită. Algoritm pentru eliminarea unui element din lista
26. Lista dublu înlanțuită. Algoritmi pentru divizarea listei și concatenarea a 2 liste
27. Coadă. Implementarea tipului abstract de date "Coadă" în limbajul C
28. Algoritmi pentru înserarea unui element în coada și ștergerea unui element din coada
29. Stiva. Implementarea tipului abstract de date "Stivă" în limbajul C
30. Algoritmi pentru înserarea unui element în stiva și ștergerea unui element din stiva
31. Arbori. Clasificarea, caracteristica generală. Parcurgerea nodurilor a unui arbore binar
32. Algoritmi iterativi de prelucrare a unui arbore binar folosind parcurgeri pe niveluri (în largime). Descrierea generală
33. Algoritmi recursivi de prelucrare a unui arbore binar folosind parcurgere recursivă. Descrierea generală
34. Algoritm iterativ de prelucrare a unui arbore binar folosind parcurgerea recursivă preordine (în adâncime). Descrierea generală

 UNIVERSITATEA TEHNICĂ A MOLDOVEI	CURRICULA UNITĂȚII DE CURS/MODULULUI	COD: F.02.O.008
	STRUCTURI DE DATE ȘI ALGORITMI	DATA: 30,05.2017 PAGINA: 13/14

35. Algoritmul iterativ pentru crearea a unui arbore binar folosind parcurgerea pe niveluri (in largime)
36. Arbore binar. Algoritmi pentru introducerea informației despre noduri și căutarea unui nod folosind parcurgere pe niveluri (in largime)
37. Arbore binar. Algoritmi de determinare a numărului de noduri și de eliberare a memoriei folosind parcurgere pe niveluri (in largime)
38. Algoritmul recursiv pentru crearea a unui arbore binar folosind parcurgerea recursivă preordine (în adâncime)
39. Arbore binar. Algoritmi recursivi pentru afișarea informații despre noduri
40. Arbore binar. Algoritmi recursivi pentru căutarea unui nod și eliberarea memoriei
41. Arbore binar. Algoritmul iterativ pentru determinarea numărului nodurilor arborelui folosind parcurgerea recursivă preordine (în adâncime)
42. Algoritmul iterativ pentru crearea a unui arbore binar folosind parcurgerea recursivă
43. preordine (în adâncime)
44. Arbore binar. Algoritmi iterativi pentru afișarea informației despre noduri și căutarea unui nod folosind parcurgerea recursivă preordine (în adâncime)
45. Arbore binar. Algoritmul iterativ de eliberare a memoriei folosind parcurgerea recursivă
46. Arbore binar de căutare și implementarea lui în limbajul C
47. Crearea arborelui binar de căutare. Algoritm pentru inserarea unui nod
48. Arbore binar de căutare . Algoritmi pentru căutarea unui nod
49. Algoritmi de căutare secvențială și binară pentru tabloul unidimensional
50. Algoritmi de sortare. Clasificarea și caracteristici generale
51. Algoritmul de sortare prin selecție
52. Algoritmi de sortare "bubble" și prin inserție
53. Algoritmi de sortare "shaker" și Shell
54. Algoritmul recursiv de sortare prin interclasare "merge"
55. Arbore binar "heap" și coada de prioritate
56. Algoritmi de sortare "coada de prioritate" și "heap"
57. Algoritmul recursiv de sortare rapidă quicksort
58. Funcții standard qsort () și bsearch () în limbajul C
59. Tehnici de programare. Clasificare și caracteristici generale
60. Tehnica de programare "Divide et empera". Exemple

VIII. REFERINȚE BIBLIOGRAFICE

1. Proiectarea Algoritmilor, [Universitea Politehnica din București, 2016, http://andrei.clubcisco.ro/cursuri/anul-2/semestrul-2/proiectarea-algoritmilor.html](http://andrei.clubcisco.ro/cursuri/anul-2/semestrul-2/proiectarea-algoritmilor.html)
2. Algoritmi si structuri de date. <https://ru.scribd.com/document/19584509/Algoritmi-Si-Structuri-de-Date>
3. Logofatu Doina, Bazele programarii in C: Aplicatii, Editura Polirom, Iasi, 2006, 407 p., Biblioteca UTM FCIM 10.ex..
4. Craus Mitica, Structuri de date si algoritmi, Iasi: Editura "Gh. Asachi", 2002, 183 p., Biblioteca UTM FCIM 2 ex..
5. Luchianova L., Stadler L., Structuri de date si algoritmi: Culegere de probleme pentru executarea lucrarilor de laborator si de control. Chisinau UTM, 2014, 32 p., Biblioteca UTM FCIM 49 ex..

 UNIVERSITATEA TEHNICĂ A MOLDOVEI	CURRICULA UNITĂȚII DE CURS/MODULULUI	COD: F.02.O.008
	STRUCTURI DE DATE ȘI ALGORITMI	DATA: 30,05.2017 PAGINA: 14/14

6. Лукьянова Л. Н., Стадлер Л. В., Структуры данных и алгоритмы: Сборник задач к выполнению лабораторных и контрольных работ, Кишинэу, ТУМ, 2010, 34 р., Biblioteca UTM FCIM 199 ex.
7. Кулев, Михаил; Лукьянова Людмила, Программирование, структуры данных и алгоритмы на языке Си : Задания для лабораторных работ. - Кишинэу: ТУМ , 2008, 28 р., Biblioteca UTM FCIM 100 ex.
8. Donald E. Knuth, The art of computer programming,
<https://www.google.com/#q=Donald,+E.+Knuth,+The+art+of+computer+programming,&start=10>