FIŞA DISCIPLINEI

Structuri de date şi algoritmi

Anul universitar 2025 – 2026

1. Date despre program

1.1. Instituţia de învăţământ superior	Universitatea Babeş-Bolyai
1.2. Facultatea	Matematică şi Informatică
1.3. Departamentul	Departamentul de Informatică
1.4. Domeniul de studii	Informatică
1.5. Ciclul de studii	Licenţă
1.6. Programul de studii / Calificarea	Informatică (în engleză)
1.7. Forma de învăţământ	Cu frecvenţă

2. Date despre disciplină

2.1. Denumirea disciplinei	**Structuri de date şi algoritmi**					Codul disciplinei	**MLE5022**
2.2. Titularul activităţilor de curs	Lect. Dr. Oneţ-Marian Zsuzsanna						
2.3. Titularul activităţilor de seminar	Lect. Dr. Oneţ-Marian Zsuzsanna						
2.4. Anul de studiu	1	2.5. Semestrul	2	2.6. Tipul de evaluare	E	2.7. Regimul disciplinei	Obligatoriu

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

3.1. Număr de ore pe săptămână	**4**	din care: 3.2. curs	**2**	3.3. seminar/ laborator/proiect	**1S + 1LP**
3.4. Total ore din planul de învăţământ	56	din care: 3.5. curs	28	3.6 seminar/laborator/proiect	**28**
Distribuţia fondului de timp pentru studiul individual (SI) şi activităţi de autoinstruire (AI)					**Ore**
Studiul după manual, suport de curs, bibliografie şi notiţe (AI)					17
Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate şi pe teren					6
Pregătire seminare/ laboratoare/ proiecte, teme, referate, portofolii şi eseuri					30
Tutoriat (consiliere profesională)					6
Examinări					10
Alte activităţi					
3.7. Total ore studiu individual (SI) şi activităţi de autoinstruire (AI)				**69**	
3.8. Total ore pe semestru				**125**	
3.9. Numărul de credite				**5**	

4. Precondiţii (acolo unde este cazul)

4.1. de curriculum	Fundamentele programării
4.2. de competenţe	Competenţe medii de programare

5. Condiţii (acolo unde este cazul)

5.1. de desfăşurare a cursului	Sală cu videoproiector
5.2. de desfăşurare a seminarului/ laboratorului	

6.1. Competenţele specifice acumulate[1]

[1] Se poate opta pentru competenţe sau pentru rezultatele învăţării, respectiv pentru ambele. În cazul în care se alege o singură variantă, se va şterge tabelul aferent celeilalte opţiuni, iar opţiunea păstrată va fi numerotată cu 6.

Competenţe profesionale/esenţiale	• dezvoltarea şi întreţinerea aplicaţiilor informatice • utilizarea bazelor teoretice ale informaticii şi a modelelor formale
Competenţe transversale	• aplicarea regulilor de muncă organizată şi eficientă, a unor atitudini responsabile faţă de domeniul didactic-ştiinţific, pentru valorificarea creativă a propriului potenţial, cu respectarea principiilor şi a normelor de etică profesională • utilizarea unor metode şi tehnici eficiente de învăţare, informare, cercetare şi dezvoltare a capacităţilor de valorificare a cunoştinţelor, de adaptare la cerinţele unei societăţi dinamice şi de comunicare în limba română şi într-o limbă de circulaţie internaţională

6.2. Rezultatele învăţării

Cunoştinţe	Studentul cunoaşte: • Absolventul are cunoştinţe necesare pentru utilizarea calculatoarelor, dezvoltarea programelor şi aplicaţiilor software, procesarea informaţiilor.
Aptitudini	Studentul este capabil să: • Absolventul este capabil să identifice probleme complexe şi să examineze probleme conexe pentru a dezvolta opţiuni de rezolvare şi implementa soluţii. • Absolventul este capabil să combine informaţii diverse pentru a formula soluţii şi genera idei de dezvoltare pentru noi produse şi aplicaţii.
Responsabilităţi şi autonomie	Studentul are capacitatea de a lucra independent pentru: • Absolventul are abilitatea de a aplica reguli generale unor probleme specifice şi de a produce soluţii relevante.

7. Obiectivele disciplinei (reieşind din grila competenţelor acumulate)

7.1 Obiectivul general al disciplinei	• Studiul structurilor de date care pot fi utilizate pentru implementarea tipurilor abstracte de date (vectori, liste înlănţuite, ansamblu, tabele de dispersie, arbori binari)
7.2 Obiectivele specifice	• Studiul conceptului de tip abstract de date şi a celor mai frecvent utilizate tipuri abstracte de date în dezvoltarea de aplicaţii. • Studiul structurilor de date care pot fi utilizate pentru implementarea acestor tipuri abstracte de date. • Dezvoltarea capacităţii de a lucra cu date stocate în diferite structuri de date şi de a compara complexitatea operaţiunilor acestora. • Dezvoltarea capacităţii de a alege structura de date adecvată pentru a modela şi rezolva probleme din lumea reală. • Dobândirea cunoştinţelor necesare pentru a lucra cu bibliotecile de structuri de date existente.

8. Conţinuturi

8.1 Curs	Metode de predare	Observaţii
1. Intorducere. Structuri de date. Tipuri abstracte de date. • Tipuri abstracte de date şi structuri de date • Convenţii de pseudocod • Complexităţi	• Expunere • Descriere • Exemple • Demonstraţie didactică	
2. Vector. Iteratori • Vector dinamic • Analiza complexităţii amortizate • Interfaţa unui iterator	• Expunere • Descriere • Conversaţia • Demonstraţie didactică	
3. Tipuri abstracte de date • TAD Mulţime: descriere, domeniu, interfaţă şi reprezentări posibile • TAD Dicţionar: descriere, domeniu, interfaţă şi reprezentări posibile • TAD Matrice: descriere, domeniu, interfaţă şi reprezentări posibile • TAD MultiDicţionar: descriere, domeniu, interfaţă şi reprezentări posibile	• Expunere • Descriere • Conversaţia • Demonstraţie didactică	
4. Tipuri abstracte de date II • TAD Stivă: descriere, domeniu, interfaţă şi reprezentări posibile • TAD Coadă: descriere, domeniu, interfaţă şi reprezentări posibile • TAD Coadă cu Priorităţi: descriere, domeniu, interfaţă şi reprezentări posibile • TAD Deque: descriere, domeniu, interfaţă şi reprezentări posibile • TAD Lista: descriere, domeniu, interfaţă şi reprezentări posibile	• Expunere • Descriere • Conversaţia • Demonstraţia didactică	
5. Liste înlănţuite • Lista simplu înlănţuită: reprezentare şi operaţii • Listă dublu înlănţuită: reprezentare şi operaţii • Iterator pentru liste înlănţuite	• Expunere • Descriere • Conversaţie • Demonstraţie didactică • Studiu de caz	
6. Liste înlănţuite II • Liste înlănţuite ordonate: reprezentare şi operaţii • Liste înlănţuite circulare: reprezentare şi operaţii • Liste înlănţuite pe tablou: reprezentare şi operaţii	• Expunere • Descriere • Conversaţie • Demonstraţie didactică	
7. Ansamblu binar • Definiţie, reprezentări, operaţii specifice • HeapSort	• Expunere • Descriere • Conversaţie • Demonstraţie didactică	
8. Tabela de dispersie • Tabela de adresare directă • Tabela de dispersie: descriere, proprietăţi • Rezolvarea coliziunilor prin liste independente	• Expunere • Descriere • Conversaţie • Demonstraţie didactică	

9. Tabela de dispersie II • Rezolvarea coliziunilor prin liste întrepătrunse • Rezolvarea coliziunilor prin adresare deschisă	• Expunere • Descriere • Conversaţie • Demonstraţie didactică	
10. Tabela de dispersie III • Dispersia perfectă • Linked hash table • Containere reprezentate pe tabele de dispersie	• Expunere • Descriere • Conversaţie • Deomnstraţie didactică	
11. Arbori. Arbori binari • Concepte legate de arbori • Aplicaţii ale arborilor • Descrierea şi proprietăţile arborilor binari • Domeniul şi interfaţa TAD Arbore binar • Reprezentări posibile pentru TAD Arbore Binar • Traversarea arborelui binar: algoritmi recursivi/non-recursivi	• Expunere • Descriere • Conversaţie • Demonstraţie didactică	
12. Arbori binari de căutare • Descriere, proprietăţi • Reprezentare • Operaţii: algoritmi recursivi şi non-recursivi • Containere reprezentare pe arbori binari de căutare	• Expunere • Descriere • Conversaţie • Demonstraţie didactică	
13. Arbore binar de căutare echilibrată • Arbori AVL	• Expunere • Descriere • Conversaţie • Demonstraţie didactică	
14. Aplicaţii ale structurilor de date studiate îm diferite limbaje de programare	• Exemplu • Expunere • Descriere • Conversaţie • Demonstraţie didactică	
Bibliografie 1. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT Press, 2009 2. S. Skiena: The algorithms design manual, Second edition, Springer, 2008 3. N. Karumanchi: Data structures and algorithms made easy, CareerMonk Publications, 2016 4. M. A. Weiss: Data structures and algorithm analysis in Java, Third Edition, Pearson, 2012 5. R. Sedgewick: Algorithms, Addison-Wesley Publishing, 1984		
8.2 laborator	Metode de predare	Observaţii
		Laboratorul este structurat sub forma a 2 ore din 2 în 2 săptămâni. Problemele de laborator atribuite la un laborator trebuie să fie prezentate în următorul laborator (cu excepţia primei teme de laborator). Fiecare laborator se concentrează pe o anumită structură de date. Studenţii vor primi un container (TAD) care trebuie să fie implementat folosind structura de date dată.
Lab 1. A1 – Vector dinamic	• Expunere	Tema se prezintă în Lab 3

• Exemplu de rezolvare a unei teme de laborator (Demo)	• Exemple • Conversaţie	
Lab 2. Discuţii despre Demo. Exemplu de operaţie extra	• Expunere • Exemple • Conversaţie	Studenţii primesc ajutor pentru rezolvarea temei A1.
Lab 3. A2 - Lista înlănţuită cu alocare dinamică	• Expunere • Exemple • Conversaţie	A1 este prezentat
Lab 4. A3 - Lista înlănţuită pe tablou	• Expunere • Exemple • Conversaţie	
Lab 5. A4 - Tabela de dispersie	• Expunere • Exemple • Conversaţie	
Lab 6. A5 - Arbore binar de căutare	• Expunere • Exemple • Conversaţie	
Lab 7. Predare problema de la Lab 6.	• Expunere • Exemple • Conversaţie	

Bibliografie:

1. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT Press, 2009
2. S. Skiena: The algorithms design manual, Second edition, Springer, 2008
3. N. Karumanchi: Data structures and algorithms made easy, CareerMonk Publications, 2016
4. M. A. Weiss: Data structures and algorithm analysis in Java, Third Edition, Pearson, 2012
5. R. Sedgewick: Algorithms, Addison-Wesley Publishing, 1984

8.3 Seminar	Metode de predare	Observaţii
		Seminarul este structurat sub forma de 2 ore din 2 în 2 săptămâni
1. TAD Colecţie cu element generic. Reprezentări şi implementări pe vector. Iterator pentru TAD Colecţie.	• Expunere • Conversaţia • Exemple • Dezbatere	
2. Complexităţi	• Expunere • Conversaţia • Exemple • Dezbatere	
3. Bucket Sort, Sortare lexicografică şi Radix Sort. Interclasarea a două liste înlănţuite sortate.	• Expunere • Conversaţia • Exemple • Dezbatere	
4. MultiDicţionar ordonat – reprezentări şi implementare pe o listă simplu înlănţuită	• Expunere • Conversaţia • Exemple • Dezbatere	
5. Evaluarea unei expresii aritmetice. Probleme rezolvate cu ansamblu binar.	• Expunere • Conversaţia • Exemple • Dezbatere	
6. Tabela de dispersie	• Expunere • Conversaţia • Exemple • Dezbatere	
7. Arbori binari	• Expunere • Conversaţia • Exemple • Dezbatere	

Bibliografie:

1. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT Press, 2009
2. S. Skiena: The algorithms design manual, Second edition, Springer, 2008
3. N. Karumanchi: Data structures and algorithms made easy, CareerMonk Publications, 2016
4. M. A. Weiss: Data structures and algorithm analysis in Java, Third Edition, Pearson, 2012
5. R. Sedgewick: Algorithms, Addison-Wesley Publishing, 1984

9. Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatori reprezentativi din domeniul aferent programului

- Conţinutul disciplinei este în concordanţă cu conţinutul cursurilor de Structuri de date de la alte universităţi din România şi din străinătate.
- Conţinutul disciplinei asigură cunoştinţele fundamentale necesare pentru utilizarea tipurilor abstracte de date şi a structurilor de date în proiectarea aplicaţiilor.

10. Evaluare

Tip activitate	10.1 Criterii de evaluare	10.2 Metode de evaluare	10.3 Pondere din nota finală
10.4 Curs	• Corectitudinea şi completitudinea cunoştinţelor asimilate. • Cunoştinţe de aplicare a conceptelor	Evaluare scrisă (în sesiune): examen scris	70%
10.5 Laborator	• Implementarea în C++ a conceptelor şi algoritmilor prezentaţi la cursuri • Documentarea lucrărilor de laborator • Respectarea termenelor de prezentare la laborator	Corectitudinea implementării şi a documentaţiei (reprezentare, specificaţii, algoritmi, complexităţi).	30%
10.6 Seminar	• Activitatea de seminar	Participare activă la discuţii în timpul seminarului	Maximum 0.5 puncte bonus adăugate la nota finală
10.7 Standard minim de performanţă			

- Cunoaşterea conceptelor de bază. Fiecare student trebuie să demonstreze că a dobândit un nivel acceptabil de cunoştinţe şi înţelegere a domeniului, că este capabil să exprime cunoştinţele dobândite într-o formă coerentă şi că are capacitatea de a utiliza aceste cunoştinţe pentru rezolvarea problemelor.
- Pentru a participa la examenul scris, un student trebuie să aibă cel puţin 6 prezenţe la laborator şi 5 prezenţe la seminar.
- Pentru a promova cu succes examenul, un student trebuie să aibă cel puţin nota 5 la examenul scris şi cel puţin nota 5 ca medie finală.

11. Etichete ODD (Obiective de Dezvoltare Durabilă / Sustainable Development Goals)[2]

Nu se aplică.

Data completării: 15.04.2025	Semnătura titularului de curs Lect. Dr. Zsuzsanna ONEȚ-MARIAN	Semnătura titularului de seminar Lect. Dr. Zsuzsanna ONEȚ-MARIAN

Data avizării în departament: ...	Semnătura directorului de departament Conf. dr. Adrian STERCA

[2] Păstrați doar etichetele care, în conformitate cu *Procedura de aplicare a etichetelor ODD în procesul academic*, se potrivesc disciplinei și ștergeți-le pe celelalte, inclusiv eticheta generală pentru *Dezvoltare durabilă* - dacă nu se aplică. Dacă nicio etichetă nu descrie disciplina, ștergeți-le pe toate și scrieți "*Nu se aplică.*".