

LEHRVERANSTALTUNGSBESCHREIBUNG

Datenstrukturen und Algorithmen

Akademisches Jahr 2025-2026

1. Angaben zum Programm

1.1. Hochschuleinrichtung	Babes-Bolyai Universität
1.2. Fakultät	Mathematik und Informatik
1.3. Department	Informatik
1.4. Fachgebiet	Informatik
1.5. Studienform	Bachelor
1.6. Studiengang / Qualifikation	Informatik
1.7. Form des Studiums	Präsenzstudium

2. Angaben zum Studienfach

2.1. LV-Bezeichnung		Datenstrukturen und Algorithmen					Code der LV	MLG5022
2.2. Lehrverantwortlicher – Vorlesung			Assoc.prof.phd. Diana Cristea					
2.3. Lehrverantwortlicher – Seminar			Assoc.prof.phd. Diana Cristea					
2.4. Studienjahr	1	2.5. Semester	2	2.6. Prüfungsform	P	2.7. Art der LV		Pflichtfach

3. Geschätzter Workload in Stunden

3.1. SWS	4	von denen: 3.2 Vorlesung	2	3.3. Seminar/Übung/Project	2
3.4. Total ore din planul de învățământ	56	von denen: 3.5 Vorlesung	28	3.6 Seminar/Übung/Project	28
Verteilung der Studienzeit:					Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften					12
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					6
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays					12
Tutoriat (consiliere profesională)					6
Prüfungen					8
Andere Tätigkeiten:					
3.7. Gesamtstundenanzahl Selbststudium			44		
3.8. Gesamtstundenanzahl / Semester			100		
3.9. Anrechnungspunkte			4		

4. Voraussetzungen (falls zutreffend)

4.1. zur Lehrveranstaltung	Grundlagen der Programmierung
4.2. kompetenzbezogene	Programmierungsfähigkeiten

5. Bedingungen (falls zutreffend)

5.1. zur Durchführung der Vorlesung	Vorlesungsraum, Beamer, Laptop
5.2. zur Durchführung des Seminars / der Übung	Labor, Beamer, Laptop

6.1. Spezifische erworbene Kompetenzen¹

¹ Man kann Kompetenzen oder Lernergebnisse, oder beides wählen. Wenn nur eine Option ausgewählt wird, wird die Tabelle für die andere Option gelöscht, und die beibehaltene Option erhält die Nummer 6.

Berufliche/Wesentliche Kompetenzen	• Nutzung der theoretischen Grundlagen der Informatik sowie formaler Modelle • Einsatz von IT-Tools in einem interdisziplinären Kontext • Entwicklung und Wartung von Softwaresystemen
Transversale Kompetenzen	• Anwendung der Regeln einer organisierten und effizienten Arbeit, einer verantwortungsvollen Haltung gegenüber dem didaktisch-wissenschaftlichen Bereich, für die kreative Nutzung des eigenen Potenzials, unter Beachtung der Grundsätze und Regeln der Berufsethik • Anwendung effizienter Methoden und Techniken zum Lernen, Informieren, Forschen und Entwickeln der Fähigkeiten, Wissen zu verwerten, sich an die Anforderungen einer dynamischen Gesellschaft anzupassen und effizient in rumänischer Sprache und in einer internationalen Sprache zu kommunizieren

6.2. Lernergebnisse

Kennt-nisse	• Der Absolvent verfügt über die notwendigen Kenntnisse zur Nutzung von Computern, zur Entwicklung von Softwareprogrammen und -anwendungen sowie zur Informationsverarbeitung.
Fähigkeiten	• Der Absolvent ist in der Lage, komplexe Probleme zu erkennen und verwandte Probleme zu untersuchen, um Lösungsmöglichkeiten zu entwickeln und Lösungen umzusetzen. • Die Absolventen sind in der Lage, verschiedene Arten von Informationen zu verknüpfen, um Lösungen zu formulieren und Ideen für die Entwicklung neuer Produkte und Anwendungen zu entwickeln.
Verantwortung und Autonomie	• Der Absolvent ist in der Lage, verschiedene Architekturen und mögliche Lösungen für ein Problem zu bewerten und auf die richtige für die spezifischen Anforderungen und Beschränkungen der zu entwickelnden Anwendung auszuwählen. • Der Absolvent ist in der Lage, allgemeine Regeln auf spezifische Probleme anzuwenden und entsprechende Lösungen zu erarbeiten.

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	• Erlernen und Verstehen der grundlegenden Datenstrukturen, sowie die Implementierung verschiedener Algorithmen.
7.2 Spezifische Ziele der Lehrveranstaltung	• Studium des Konzeptes abstrakter Datentyp, sowie der am meisten verwendeten abstrakte Datentypen • Studium der passenden Datenstrukturen für die Implementierung abstrakter Datentypen (Arrays, verkettete Listen, Heaps, Hashtabellen, Binärbäume, usw.) • Die Fähigkeit einen bestimmten Datentyp für die Lösung eines Problems auszuwählen und Anwendungen zu entwerfen die verschiedene abstrakte Datentypen verwenden. • Die Fähigkeit Daten die in verschiedene Datenstrukturen gespeichert sind zu verwenden. • Die Fähigkeit Komplexitäten unterschiedlichen Algorithmen zu vergleichen

8. Inhalt

8.1 Vorlesung	Lehr-und Lernmethode	Anmerkungen
1. Einführung. Datenstrukturen. Abstrakte Datentypen: • Abstrahierung • Komplexitäten • Pseudocode Notationen	• Darstellung der Thematik • Diskussion • Vortrag	
2. Arrays. Iteratoren. Amortisierte Analyse	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
3. Abstrakte Datentypen:- Beschreibung, Domäne, Schnittstelle/Interface, Repräsentierung: • ADT Set/Menge & sortierte Menge • ADT Bag/Multi-Menge & sortierte Multi-Menge • ADT Matrix	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
4. Abstrakte Datentypen: Beschreibung, Domäne, Schnittstelle und mögliche Repräsentierungen • ADT Queue/Schlange • ADT Prioritätsschlange • ADT Stack/Stapel • ADT Map & sortierte Map • ADT MultiMap	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
5. Abstrakte Datentypen: Beschreibung, Domäne, Schnittstelle und mögliche Repräsentierungen • ADT List • Verkettete Listen: • Einfach verkettete Listen: Repräsentierung und Operationen • Doppelt verkettete Listen: Repräsentierung und Operationen	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
6. Abstrakte Datentypen: • Zykläre Listen: Repräsentierung und Operationen • Sortierte verkettete Listen: Repräsentierung und Operationen • Verkettete Listen auf Arrays: Repräsentierung und Operationen	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
7. Andere mögliche Repräsentierungen und Aufgaben mit ADT Stack, Queue, PriorityQueue ADT Dequeue: Beschreibung und mögliche Repräsentierungen	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	

Skip Listen		
8. Binären Heap: • Definition, Repräsentierung, spezifische Operationen • HeapSort • Prioritätsschlange: Repräsentierung auf Heaps ADT Binomial-Heap: • Beschreibung, Repräsentierung und Operationen	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
9. Hashtabellen: • Beschreibung und Eigenschaften • Kollisionbehandlung mittels Verkettung • Kollisionbehandlung mittels offener Adressierung	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
10. Hashtabellen: • Kollisionbehandlung mittels coalesced Hashing • Perfektes Hashing • Cuckoo Hashing • Hopscotch Hashing • Verkettete Hashtabelle	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
11. Bäume: • Beschreibung, Eigenschaften • Anwendungen Binärbäume: • Beschreibung, Eigenschaften • Domäne und Schnittstelle für ADT Binärbaum • Operationen • Traversierungen: rekursive und nicht-rekursive Algorithmen	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
12. Binärbäume: • Traversierungen: rekursive und nicht-rekursive Algorithmen Binärsuchbäume: • Beschreibung, Eigenschaften • Repräsentierung • Operationen: rekursive und nicht-rekursive Algorithmen • Containers repräsentiert auf Binärsuchbäume	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch • Diskussion • Beispiele	
13 Balancierte Binärsuchbäume: AVL Bäume	• Vortrag • Beschreibung • Erklärungen • Unterrichtsgespräch	

	• Diskussion • Beispiele	
14 Balancierte Binärsuchbäume: • AVL Bäume Aufgaben mit den abstrakten Datentypen (Infix- und Postfixnotation, Klammerproblem, Rot-Schwarz Kartenspiel, usw.)	• Vortrag • Beschreibung • Erklärungen • Diskussion • Beispiele	
Literatur In deutscher Sprache: 1. Dietzfelbinger, M., Mehlhorn, K., Sanders, P., Algorithmen und Datenstrukturen. Die Grundwerkzeuge, Springer, 2014 2. Weicker, K., Weicker, N., Algorithmen und Datenstrukturen, Springer, 2013 3. Sedgewick, R., Algorithmen, Addison Wesley, New York, 1991. 4. Wirth, N., Datenstrukturen und Algorithmen, Teubner, Stuttgart, 1999. 5. Solymosi, A., Grude, U., Grundkurs Algorithmen und Datenstrukturen in Java, Eine Einführung in die praktische Informatik, Vieweg Teubner, Wiesbaden, 2008. 6. Ottmann, Th., Widmayer, P., Algorithmen und Datenstrukturen, Springer, 1997. 7. Sattler, K-U., Algorithmen und Datenstrukturen, Springer, 2008. In anderen Sprachen: 8. Niculescu V., Czibula G., Structuri fundamentale de date. O perspective orientate obiect. Editura Casa Cartii de Stiinta, Cluj-Napoca, 2011 9. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT 10. Horowitz, E.: Fundamentals of Data Structures in C++. Computer Science Press, 1995. 11. Frentiu M., Pop H.F., Serban G., Programming Fundamentals, Ed.Presa Universitara Clujeana, Cluj-Napoca, 2006, 234 pagini		
8.2 Seminar / Laborarbeit	Lehr-und Lernmethode	Anmerkungen
Seminar		Seminar - 2 Stunden alle 2 Wochen
S1. ADT Bag/MultiMenge – Repräsentierung und Implementation auf Array. Iterator für ADT Bag	Beschreibung, Erklärungen, Diskussionen, Beispiele, Übungen	
S2. Komplexitäten	Beschreibung, Erklärungen, Diskussionen, Beispiele, Übungen	
S3. Sortierte MultiMap – Repräsentierung und Implementation auf verkettete Listen	Beschreibung, Erklärungen, Diskussionen, Beispiele, Übungen	
S4. BucketSort, Lexikographischer Sort, Radix Sort. Merge von sortierten Listen Indexierte Listen	Beschreibung, Erklärungen, Diskussionen, Beispiele, Übungen	
S5. Schriftlicher Test und Übungen	Schriftlicher Test	1 Stunde
	Beschreibung, Erklärungen, Diskussionen, Beispiele, Übungen	1 Stunde
S6. Hashtabellen. Kollisionbehandlung mittels coalesced Hashing	Beschreibung, Erklärungen, Diskussionen, Beispiele, Übungen	

S7. Binärbäume	Beschreibung, Erklärungen, Diskussionen, Beispiele, Übungen	
Labor		Labor - 2 Stunden alle 2 Wochen Jede Aufgabe muss in der nächsten Woche abgegeben werden. Ein Labor basiert auf eine Datenstruktur. Die Aufgaben bestehen aus unterschiedlichen Containers die mit Hilfe der gegebenen Datenstruktur implementiert werden müssen.
L1-L2. Dynamisches Array	Unterrichtsgespräch, Erklärungen, Beispiele	Für Alle Aufgaben: • Implementierung • Dokumentation • Komplexitäten
L3. Einfach verkettete Liste mit dynamischer Allokation	Unterrichtsgespräch, Erklärungen, Beispiele	Für Alle Aufgaben: • Implementierung • Dokumentation • Komplexitäten
L4. Einfach verkettete Liste auf Arrays	Unterrichtsgespräch, Erklärungen, Beispiele	Für Alle Aufgaben: • Implementierung • Dokumentation • Komplexitäten
L5. Hashtabelle	Unterrichtsgespräch, Erklärungen, Beispiele	Für Alle Aufgaben: • Implementierung • Dokumentation • Komplexitäten
L6.-L7. Binärsuchbäume	Unterrichtsgespräch, Erklärungen, Beispiele	Für Alle Aufgaben: • Implementierung • Dokumentation • Komplexitäten

Literatur

In deutscher Sprache:

1. Dietzfelbinger, M., Mehlhorn, K., Sanders, P., Algorithmen und Datenstrukturen. Die Grundwerkzeuge, Springer, 2014
2. Weicker, K., Weicker, N., Algorithmen und Datenstrukturen, Springer, 2013
3. Sedgewick, R., Algorithmen, Addison Wesley, New York, 1991.
4. Wirth, N., Datenstrukturen und Algorithmen, Teubner, Stuttgart, 1999.
5. Solymosi, A., Grude, U., Grundkurs Algorithmen und Datenstrukturen in Java, Eine Einführung in die praktische Informatik, Vieweg Teubner, Wiesbaden, 2008.
6. Ottmann, Th., Widmayer, P., Algorithmen und Datenstrukturen, Springer, 1997.
7. Sattler, K-U., Algorithmen und Datenstrukturen, Springer, 2008.

In anderen Sprachen:

8. Niculescu V., Czibula G., Structuri fundamentale de date. O perspectiva orientata obiect. Editura Casa Cartii de Stiinta, Cluj-Napoca, 2011
9. T. Cormen, C. Leiserson, R. Rivest, C. Stein: Introduction to algorithms, Third Edition, The MIT
10. Horowitz, E.: Fundamentals of Data Structures in C++. Computer Science Press, 1995.
11. Frentiu M., Pop H.F., Serban G., Programming Fundamentals, Ed.Presa Universitara Clujeana, Cluj-Napoca, 2006, 234 pagini

9. Verbindung der Inhalte mit den Erwartungen der Wissensgemeinschaft, der Berufsverbände und der für den Fachbereich repräsentativen Arbeitgeber

- Der Kurs folgt die IEEE und ACM Curricula Empfehlungen für das Informatikstudium.
- Der Kurs konzentriert sich auf die Lösung der Probleme die ein Absolvent an seiner zukünftigen Arbeitsstelle lösen muss.
- Der Kurs existiert in der Mehrzahl der rumänischen und ausländischen Universitäten.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	• Korrekter Umgang mit den Grundbegriffen der Vorlesung. • Kenntnisse für das Lösen von Aufgaben	Schriftliche Prüfung (in der Prüfungszeit)	60%
10.5 Übung	• Korrektheit und Effizienz in der gelösten Aufgaben • Dokumentation • Deadlines einhalten	Praktische Aufgaben	30%
10.5 Seminar	• Korrekter Umgang mit den Grundbegriffen der Vorlesung. • Kenntnisse für das Lösen von Aufgaben	Schriftlicher Test (Mitte des Semesters)	10%
10.6 Minimale Leistungsstandards			

Um in die Prüfung eingelassen zu werden gelten folgende Kriterien:

- Wenigstens 6 Anwesenheiten bei den Übungen/Labor
- Wenigstens 5 Anwesenheiten bei dem Seminar
- Bei den praktischen Aufgaben muss die Mindestnote 5 erzielt werden

Für das Bestehen der Prüfung gelten folgende Kriterien:

- bei der schriftlichen Prüfung muss die Mindestnote 5 erzielt werden, ohne Aufrundung (nur die Endnote wird auf ganze Zahl gerundet)
- bei den praktischen Aufgaben muss die Mindestnote 5 erzielt werden, ohne Aufrundung
- die Endnote muss mindestens 5 sein

11. SDD-Nachhaltigkeits-Logos (Sustainable Development Goals)²

Nicht anwendbar.

Ausgefüllt am:

21.04.2025

Vorlesungsverantwortlicher

Assoc.prof.phd. Diana Cristea

Seminarverantwortlicher

Assoc.prof.phd. Diana Cristea

Genehmigt im Department am:

Departmentleiter/in

Assoc.prof.phd. Adrian STERCA

² Bitte belassen Sie nur die Logos, die entsprechend den [*Regularien zu Anwendung der Nachhaltigkeits-Logos im akademischen Betrieb*](#) dem jeweiligen Studienfach entsprechen und löschen Sie diejenigen Logos, inklusive das allgemeine *Nachhaltigkeits-Logo* falls dieses nicht zutrifft. Falls keines der Logos für das Studienfach anwendbar ist, löschen Sie alle mit der Angabe „*nicht anwendbar*“.