

LEHRVERANSTALTUNGSBESCHREIBUNG

Fortgeschrittene Programmierungsmethoden

2 Akademisches Jahr 2025-2026

1. Angaben zum Programm

1.1. Hochschuleinrichtung	Babeş-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 in deutscher Sprache
1.7. Form des Studiums	IF

2. Angaben zum Studienfach

2.1. LV-Bezeichnung		Fortgeschrittene Programmierungsmethoden					Code der LV	MLG5008
2.2. Lehrverantwortlicher – Vorlesung			Lect. dr. Cătălin Rusu					
2.3. Lehrverantwortlicher – Seminar			Lect. dr. Cătălin Rusu					
2.4. Studienjahr	2	2.5. Semester	3	2.6. Prüfungsform	P	2.7. Art der LV		Pflichtfach

3. Geschätzter Workload in Stunden

3.1. SWS	5	von denen: 3.2 Vorlesung	2	3.3. Seminar/Übung	1+2
3.4 Gesamte Stundenanzahl im Lehrplan	70	von denen: 3.5 Vorlesung	28	3.6 Seminar/Übung	14+28
Verteilung der Studienzeit:					Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften					20
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					10
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays (mai mare sau egal cu nr. total ore prevăzut în calendarul disciplinei pentru temele de control)					30
Tutoriat (consiliere profesională)					10
Prüfungen					10
Andere Tätigkeiten: [de ex.: comunicare bidirecțională cu titularul de disciplină / tutorele]					-
3.7. Gesamtstundenanzahl Selbststudium			80		
3.8. Gesamtstundenanzahl / Semester			150		
3.9. Anrechnungspunkte			6		

4. Voraussetzungen (falls zutreffend)

4.1. zur Lehrveranstaltung	- Objektorientierte Programmierung - Datenstrukturen und Algorithmen
4.2. kompetenzbezogene	Grundlegende Begriffe der objektorientierter Programmierung.

5. Bedingungen (falls zutreffend)

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

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	• K1.1 Geeignete Beschreibung der Paradigmen der Programmierung und der spezifischen Sprachmechanismen, sowie die Identifizierung der Differenzen zwischen semantischen und syntaktischen Aspekten • K1.2 Erklärung existierender Softwareanwendungen auf verschiedenen Niveaus (Architektur, Pakete, Klassen, Methoden), anhand geeigneter Anwendung der Grundkenntnisse • K1.3 Entwickeln von geeigneten Quellcodes und unitäres Testen von Komponenten in einer bekannten Programmiersprache, anhand gegebener Entwurfsspezifikationen • K1.4 Testen der Anwendungen anhand von Testplänen • K1.5 Entwurf von Programmeinheiten und Verfassung der geeigneten Dokumentationen
Transversale Kompetenzen	• TK1 Anwendung der Regeln für gut organisierte und effiziente Arbeit, für verantwortungsvolle Einstellungen gegenüber der Didaktik und der Wissenschaft, für kreative Förderung des eigenen Potentials, mit Rücksicht auf die Prinzipien und Normen der professionellen Ethik • TK3 Anwendung von effizienten Methoden und Techniken für Lernen, Informieren und Recherchieren, für das Entwickeln der Kapazitäten der praktischen Umsetzung der Kenntnisse, der Anpassung an die Bedürfnisse einer dynamischen Gesellschaft, der Kommunikation in rumänischer Sprache und in einer internationalen Verkehrssprache

6.2. Lernergebnisse

Kennt-nisse	Der/die Studierende kennt: ...
Fähigkeiten	Der/die Studierende ist in der Lage...
Verantwortung und Autonomie	Der/die Studierende verfügt über die Fähigkeit, selbstständig zu arbeiten, um ...

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	• Kenntnisse im Umgang mit Datenmodellen, insbesondere die der objektorientierter Programmierung • Anwenden der Schablone • Anwenden von frameworks
7.2 Spezifische Ziele der Lehrveranstaltung	• Analyse und Design von Softwaresysteme • Anlernen grundlegender Kenntnisse über UML • Aneignen von Schablonen. • Java • C#

8. Inhalt

8.1 Vorlesung	Lehr- und Lernmethode	Anmerkungen
1. Einführung in Java I	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
2. Einführung in Java II	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
3. IO. Collections. Generische Typen	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
4. Funktionale Programmierung in JAVA. JAVA8. Streams	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
5. GUI. JAVA FX.	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
6. Treads.	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
7-8. XML: schema, documents. JSON	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
9. GUI (Teil 2): FXML, CSS. Metaprogramming: reflection, serialization	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
10. Webprogrammierung. JSP	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
11. Einführung in C#	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
12. LINQ	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
13-14. GRASP und GoF Schablone. Das objektorientierte Design. UML	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
Literatur - James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition. 2. Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006 E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994 - Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilley, 2010 ***, Microsoft Developer Network, Microsoft Inc., http://msdn.microsoft.com/ 6. ***, The Java Tutorial, 2013. http://download.oracle.com/javase/tutorial/ Auf Deutsch: - Gregor Fischer, Jurgen Wolff von Gudenberg, Programmieren in Java, Springer, Berlin, Heidelberg, New York, 2005. 2. Detlef Sesse, Grundkurs Programmieren in Java 1, Hanser Verlag, Muenchen, Wien, 2007.		
8.2 Seminar /Übung	Lehr-und Lernmethode	Anmerkungen
1. Einfache Übungen	Beispiele, Diskussionen, Debatte, Erklärungen	
2. Java Projekt: Collections, Generics	Beispiele, Diskussionen, Debatte, Erklärungen	
3. Java Projekt: Generics	Beispiele, Diskussionen, Debatte, Erklärungen	

4. Java Projekt: IO	Beispiele, Diskussionen, Debatte, Erklärungen	
5. Java Projekt: Functional programming	Beispiele, Diskussionen, Debatte, Erklärungen	
6. Java Projekt: GUI	Beispiele, Diskussionen, Debatte, Erklärungen	
7. Java Projekt: concurrency	Beispiele, Diskussionen, Debatte, Erklärungen	
8. C# Projekt	Beispiele, Diskussionen, Debatte, Erklärungen	

Literatur in deutscher Sprache

- James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. The Java™ Language Specification Java SE 7 Edition
- Eckel, B., Thinking in Java, 4th edition, Prentice Hall, 2006
- E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994
- Joseph Albahari and Ben Albahari, C# 4.0 in a Nutshell, Fourth Edition, O'Reilly, 2010
- Microsoft Developer Network, Microsoft Inc., <http://msdn.microsoft.com/>
- The Java Tutorial, 2013. <http://download.oracle.com/javase/tutorial/>

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

- Diese Vorlesung wird an international bekannten Universitäten im Fachgebiet Informatik angeboten.
- Der Inhalt der Vorlesung ist wichtig für die Softwarefirmen und entspricht der ACM Richtlinien.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	Angeeignete Kenntnisse	schriftliche Abschlussarbeit	40%
10.5 Seminar/Übung	Programmieren	Praktischer Test	30%
	Laborarbeiten	Dokumentation, Diskussion	30%
10.6 Minimale Leistungsstandards			
• Note jeder Übung (Labor) soll größer als 5 sein. Für das Bestehen der praktischen Prüfung muss die Mindestnote 5 erreicht werden. Für das Bestehen der schriftlichen Prüfung muss die Mindestnote 5 erzielt werden. • Nur die Endnote wird auf ganze Zahl gerundet. • Erforderliche Anwesenheit beim: Seminar: 75%; Labor: 90%.			

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

	Allgemeines Logo für die SDG-Initiative							
								

Ausgefüllt am:
11.04.2025

Vorlesungsverantwortlicher

Lect. dr. Cătălin Rusu

Seminarverantwortlicher

Lect. dr. Cătălin Rusu

Genehmigt im Department am:
25.04.2025

Departmentleiter

Conf. Dr. 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“.