

LEHRVERANSTALTUNGSBESCHREIBUNG

Software Projekt

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		Software Projekt					Code der LV	MLG5012
2.2. Lehrverantwortlicher – Vorlesung			Kovacs Karoly					
2.3. Lehrverantwortlicher – Seminar			Kovacs Karoly					
2.4. Studienjahr	3	2.5. Semester	5	2.6. Prüfungsform	C	2.7. Art der LV	Wahlpflichtfach	

3. Geschätzter Workload in Stunden

3.1. SWS	2	von denen: 3.2 Vorlesung	0	3.3. Seminar/Übung	2
3.4 Gesamte Stundenanzahl im Lehrplan	28	von denen: 3.5 Vorlesung	0	3.6 Seminar/Übung	28
Verteilung der Studienzeit:					Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften					5
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					5
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays					5
Tutoriat (consiliere profesională)					5
Prüfungen					2
Andere Tätigkeiten:					-
3.7. Gesamtstundenanzahl Selbststudium		22			
3.8. Gesamtstundenanzahl / Semester		50			
3.9. Anrechnungspunkte		2			

4. Voraussetzungen (falls zutreffend)

4.1. zur Lehrveranstaltung	-
4.2. kompetenzbezogene	Höhere Programmiersprachen Kompetenzen

5. Bedingungen (falls zutreffend)

5.1. zur Durchführung der Vorlesung	· Vorlesungsraum, Beamer, Laptop
5.2. zur Durchführung des Seminars / der Übung	· Computerraum besorgt mit Entwicklungsumgebungen

6.1. Spezifische erworbene Kompetenzen¹

Berufliche/Wesentliche Kompetenzen	• K2.1 Identifizierung geeigneter Methoden für die Entwicklung von Softwaresystemen • K2.2 Identifizierung und Erklärung geeigneter Mechanismen für die Spezifizierung von Softwaresystemen • K2.3 Benutzung der Methoden, Spezifizierungsmechanismen und Entwurfsmedien für die Entwicklung von Software-Anwendungen • K2.4 Benutzung von geeigneten Kriterien und Methoden für die Auswertung von Software-Anwendungen • K2.5 Entwurf von spezifischen Software-Anwendungen
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 • TK2 Effiziente Durchführung der in einer interdisziplinären Gruppe organisierten Aktivitäten und Entwicklung empathischer Fähigkeiten zur zwischenmenschlichen Kommunikation, zum Aufbau von Beziehungen und zur Zusammenarbeit mit verschiedenen Gruppen • 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 Student kennt: • die Methoden, Algorithmen, Paradigmen und Techniken, die in verschiedenen Bereichen der Informatik verwendet werden. • die Nutzung von Computern, die Entwicklung von Programmen und Softwareanwendungen sowie die Informationsverarbeitung.
Fähigkeiten	Der Student ist in der Lage: • die Methoden, Algorithmen, Paradigmen und Techniken, die in verschiedenen Bereichen der Informatik verwendet werden, zu präsentieren und zu erklären. • Programmierparadigmen (prozedural, objektorientiert, funktional) zur Entwicklung von Softwareanwendungen einzusetzen, die den spezifischen Anforderungen des jeweiligen Fachgebiets entsprechen. • Informationen zu verstehen und effektiv zu kommunizieren.
Verantwortung und Autonomie	Der Student ist in der Lage, selbstständig zu arbeiten, um: • neue Anwendungen, Systeme oder Produkte zu entwickeln, zu entwerfen und zu erstellen, unter Anwendung bewährter Praktiken aus dem Fachbereich. • Computerprogramme zu entwerfen und Softwaresysteme zu analysieren.

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	• Teamwork Fähigkeiten • Software Projekt Management
--	---

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.

7.2 Spezifische Ziele der Lehrveranstaltung	• Modellorientierte Entwicklung • Testorientierte Entwicklung
--	--

8. Inhalt

8.1 Vorlesung	Lehr-und Lernmethode	Anmerkungen
8.2 Seminar / Übung	Lehr-und Lernmethode	Anmerkungen
1. Agile Model Driven Development (AMDD)	Beispiele, Diskussionen, Teamarbeit	
2. Open Unified Process (OpenUP)	Beispiele, Diskussionen, Teamarbeit	
3. Version control systems	Beispiele, Diskussionen, Teamarbeit	
4. Tracking systems	Beispiele, Diskussionen, Teamarbeit	
5. Requirements modeling	Beispiele, Diskussionen, Teamarbeit	
6. Initial domain model	Beispiele, Diskussionen, Teamarbeit	
7. Initial architectural modeling	Beispiele, Diskussionen, Teamarbeit	
8. Projektplanung	Beispiele, Diskussionen, Teamarbeit	
9. Iteration Modellierung	Beispiele, Diskussionen, Teamarbeit	
10. Iteration Modellierung	Beispiele, Diskussionen, Teamarbeit	
11. Test-driven development	Beispiele, Diskussionen, Teamarbeit	
12. Test-driven development	Beispiele, Diskussionen, Teamarbeit	
13. Evaluation criteria for the project milestones	Beispiele, Diskussionen, Teamarbeit	
14. Abgabe	Diskussionen	
Literatur 1. Bugzilla, http://www.bugzilla.org/ 2. OpenUP, http://epf.eclipse.org/wikis/openup/ 3. Scott W. Ambler. Agile Model Driven Development (AMDD): The Key to Scaling Agile Software Development. http://www.agilemodeling.com/essays/amdd.htm 4. Subversion, http://subversion.tigris.org/ 5. Meier, Harald. "Internationales Projektmanagement." (2015). 6. Herrmann, A., Knauss, E., Fahney, R., Gartung, T., Glunde, J., Hoffmann, A., & Valentini, U. (2013). Requirements engineering und Projektmanagement. R. Weißbach (Ed.). Springer Vieweg. http://www.swi-prolog.org/		

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 des Kurses gilt als wichtiger Teil der Programmierkenntnisse der Informatiker in Software Unternehmen.
--

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung			
10.5 Seminar / Übung	Projekte	Vorstellung der Projekte	100%
10.6 Minimale Leistungsstandards			
Die Gesamtnote min 5 auf einer Skala von 1 bis 10.			

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

	Allgemeines Logo für die SDG-Initiative							

Ausgefüllt am:
11.04.2025

Vorlesungsverantwortlicher
Kovacs Karoly

Seminarverantwortlicher
Kovacs Karoly

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“.