

LEHRVERANSTALTUNGSBESCHREIBUNG

Softwaresystemtechnik

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		Softwaresystemtechnik					Code der LV	MLG5011
2.2. Lehrverantwortlicher – Vorlesung			Conf. Dr. Christian Bartelt					
2.3. Lehrverantwortlicher – Seminar			Conf. Dr. Christian Bartelt					
2.4. Studienjahr	2	2.5. Semester	2	2.6. Prüfungsform	E	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	2
3.4 Gesamte Stundenanzahl im Lehrplan	56	von denen: 3.5 Vorlesung	28	3.6 Seminar/Übung	28
Verteilung der Studienzeit:					Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften					14
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					14
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und					38
Tutoriat (consiliere profesională)					14
Prüfungen					14
Andere Tätigkeiten:					-
3.7. Gesamtstundenanzahl Selbststudium			94		
3.8. Gesamtstundenanzahl / Semester			150		
3.9. Anrechnungspunkte			6		

4. Voraussetzungen (falls zutreffend)

4.1. zur Lehrveranstaltung	- Grundlagen der Programmierung. Objektorientierte Programmierung
4.2. kompetenzbezogene	-

5. Bedingungen (falls zutreffend)

5.1. zur Durchführung der Vorlesung	Videoprojektor, Internetzugang
5.2. zur Durchführung des Seminars / der Übung	Computers, UML Weekzeuge

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	• 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 Effizienter Ablauf der Tätigkeiten in einer interdisziplinären Gruppe, das Entwickeln der Kapazitäten für empathische zwischenmenschliche Kommunikation, Verknüpfung und Zusammenarbeit mit unterschiedlichen 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 Absolvent verfügt über die notwendigen Kenntnisse zu den Phasen des Software-Lebenszyklus und den Modellen von Softwareprozessen. • Der Absolvent besitzt die notwendigen Kenntnisse zur Anwendung modellbasierter Softwareentwicklungstechniken. • Der Absolvent hat das nötige Wissen über die UML-Sprache sowie die Fähigkeit, CASE-Tools zu nutzen, um Softwaresysteme zu verstehen, zu dokumentieren und zu implementieren.
Fähigkeiten	• Der Absolvent verfügt über die nötigen Fähigkeiten zur Entwicklung von Computerprogrammen und zur Analyse von Softwaresystemen. • Der Absolvent besitzt die erforderlichen Fähigkeiten zum Verständnis und zur Anwendung objektorientierter Programmierkonzepte bei der Entwicklung von Softwareanwendungen mittlerer bis hoher Komplexität.
Verantwortung und Autonomie	• Der Absolvent ist in der Lage, verschiedene Informationen zu kombinieren, um Lösungen zu formulieren und neue Produkt- und Anwendungsideen zu entwickeln. • Der Absolvent ist fähig, funktionale und nicht-funktionale Anforderungen, wie sie in spezifischen Dokumenten zur Softwareanalyse und -entwurf beschrieben sind, umzusetzen. • Der Absolvent kann Architektur- und Entwurfsmuster sowie bewährte Praktiken des Fachgebiets anwenden, um Softwareanwendungen mittlerer bis hoher Komplexität zu entwerfen.

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	• das Aneignen der Kenntnisse zur Softwareentwicklung
7.2 Spezifische Ziele der Lehrveranstaltung	• das Aneignen der Kenntnisse über Methodologien der Softwareentwicklung • die Vertrautheit mit Konzepten der modernen Softwareentwicklung • das Verständnis der Wichtigkeit von Softwaredokumentation in alle Etappe des Lebenszyklus

8. Inhalt

8.1 Vorlesung	Lehr-und Lernmethode	Anmerkungen
1. Organisatorische Diskussionen. Lebenszyklus eines Softwareprodukts. Softwaretechnik: Evolution. Definitionen. Das Wasserfallmodell. Das Spiralmodell. Inkrementelles Vorgehensmodell. Extreme Programming	Vortrag, Gespräch, Fallstudien	
2. Problemlösen und Softwareentwicklung. Prinzipien der Kommunikation.	Vortrag, Gespräch, Fallstudien	
3. Systemmodellierung. Anwendungsdomäne. Werkzeuge zur Systemmodellierung	Vortrag, Gespräch, Fallstudien	
4. Überblick über UML. Anforderungsanalyse. Modellierung von Klassen.	Vortrag, Gespräch, Fallstudien	
5. UML. Dynamische Modellierung. Architekturmodellierung.	Vortrag, Gespräch, Fallstudien	
6. Anforderungsanalyse	Vortrag, Gespräch, Fallstudien	
7. Statische Modellierung. Dynamische Modellierung. Grundlagen der Modellierung. Entwurfsetappen	Vortrag, Gespräch, Fallstudien	
8. Architektur und Abhängigkeitsbeziehungen. Prinzipien der Architekturmodellierung	Vortrag, Gespräch, Fallstudien	
9. Architekturmodellierung. Datenmodellierung	Vortrag, Gespräch, Fallstudien	
10. Die logische Systemarchitektur. Etappen der objektorientierte Modellierung	Vortrag, Gespräch, Fallstudien	
11. Interfacedesign	Vortrag, Gespräch, Fallstudien	
12. Mensch-Computer-Interaktion	Vortrag, Gespräch, Fallstudien	
13. Methoden beim Testen von Software	Vortrag, Gespräch, Fallstudien	
14. Der Faktor Mensch in der Softwareentwicklung	Vortrag, Gespräch, Fallstudien	
Literatur in deutscher Sprache • GRUHN, V., PIEPER, D., ROTTGERS, C., Effektives Software Engineering mit UML2 und Eclipse, Springer 2006. • RUMPE, B., Agile Modellierung mit UML: Codegenerierung, Testfälle, Refactoring, Springer, 2005. • KLEUKER, ST., Grundkurs Software Engineering mit UML, Vieweg Teubner, 2011. • Sommerville, Software Engineering, Pearson Studium IT, 2015 • J. Ludewig, H. Lichter, Software Engineering: Grundlagen, Menschen, Prozesse, Techniken, dpunkt Verlag, 2017		
Sonstige Literatur • IACOBSON, I., BOOCH, G., RUMBAUGH, J.: The Unified Software Development Process, Addison-Wesley, 1999. • MARTIN, R.C.: Agile Software Development: Principles, Patterns, and Practices, Prentice Hall, 2002. • PÂRV, B.: Analiza si proiectarea sistemelor, Univ. Babes-Bolyai, CFCID, Facultatea de Matematica si Informatica, Cluj-Napoca, 2002, 2003, 2004. • PRESSMAN, R.S.: Software Engineering - A Practitioners Approach, McGraw-Hill, 3rd ed. 1992; 4th ed. 1996, 5th ed. 2001, 6th ed. 2005. • SCHACH, S.R.: Object-Oriented and Classical Software Engineering, McGraw-Hill, 5th ed., 2002, 6th ed. 2005.		
8.2 Seminar /Übung	Lehr-und Lernmethode	Anmerkungen
S1. Fallstudie: Anforderungsanalyse (1) - Grundanforderungen	Rückschau, Erklärungen, Beispiele	
S2. Fallstudie: Anforderungsanalyse (2) - modellierung von Klassen	Rückschau, Erklärungen, Beispiele	

S3. Prüfung: Anforderungsanalyse (3) – Anwendungsfall-Modellierung. Sequenzdiagramme .	Rückschau, Erklärungen, Beispiele	
S4. Prüfung 1: Objektorientierte Modellierung	Rückschau, Erklärungen, Beispiele	
S5. Fallstudie: Entwurf (1) - Anwendungsfälle		
S6. Prüfung 2: Entwurf (2). GRASP		
S7. Fallstudien		
L1. Organisatorische Diskussionen	Erklärungen, Beispiele	2 Stunden jeden 2 Wochen
L2. Start des Projekts (Etappe 1)	Erklärungen, Beispiele	
L3. Projekt (Etappe 1) Bildung	Erklärungen, Beispiele	
L4. Projekt (Etappe 1) abgeben. Start des Projekts (Etappe 2)	Erklärungen, Beispiele	
L5. Projekt (Etappe 2) Bildung	Erklärungen, Beispiele	
L6. Projekt (Etappe 2) abgeben. Start des Projekts (Etappe 3)	Erklärungen, Beispiele	
L7. Projekt (Etappe 3) abgeben	Erklärungen, Beispiele	
Literatur • Object Mentor Homepage, [http://www.objectmentor.com]. • Agile Modeling Homepage, [http://www.agilemodeling.com]. • Software Engineering Body of Knowledge, IEEE, 2004. [http://www.swebok.org].		

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 existiert in der Mehrzahl der rumänischen und ausländischen Universitäten.
- Die Softwarefirmen finden den Kursinhalt zehr wichtig für die Ausbildung der Zukünftigen Softwareentwickler.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	Kenntnisse der im Kurs behandelten Themen	2 Tests Midterm-Prüfung Prüfung	10% 20% 25%
10.5 Seminar / Übung	Die Fähigkeit Modellierungstechniken für das Lösen konkreter Probleme einzusetzen	3 Mini-projekte der Besuch des Kurses	45% 10%
10.6 Minimale Leistungsstandards			
Note 5 auf einer Skala von 1 bis 10.			

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

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“.

	Allgemeines Logo für die SDG-Initiative							

Ausgefüllt am:
11.04.2025

Vorlesungsverantwortlicher

Ponf.Dr. Christian Bartelt

Seminarverantwortlicher

Ponf.Dr. Christian Bartelt

Genehmigt im Department am:
25.04.2025

Departmentleiter

Conf. Dr. Adrian Sterca