

LEHRVERANSTALTUNGSBESCHREIBUNG

Programmierung für mobile Geräte

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	Vollzeit

2. Angaben zum Studienfach

2.1. LV-Bezeichnung		Programmierung für mobile Geräte					Code der LV	MLG5078
2.2. Lehrverantwortlicher – Vorlesung								
2.3. Lehrverantwortlicher – Seminar								
2.4. Studienjahr	3	2.5. Semester	5	2.6. Prüfungsform	E	2.7. Art der LV	Pflichtfach	

3. Geschätzter Workload in Stunden

3.1. SWS	3	von denen: 3.2 Vorlesung	2	3.3. Seminar/Übung/Project	1
3.4. Gesamte Stundenanzahl im Lehrplan	42	von denen: 3.5 Vorlesung	28	3.6 Seminar/Übung/Project	14
Verteilung der Studienzeit:					Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften					12
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					12
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays					12
Tutoriat (consiliere profesională)					10
Prüfungen					12
Andere Tätigkeiten:					
3.7. Gesamtstundenanzahl Selbststudium		58			
3.8. Gesamtstundenanzahl / Semester		100			
3.9. Anrechnungspunkte		4			

4. Voraussetzungen (falls zutreffend)

4.1. zur Lehrveranstaltung	Algorithmus, Datenstrukturen
4.2. kompetenzbezogene	Fähigkeiten in Programmierungsumgebungen in Hochsprache (objektorientiert), vorzugsweise JAVA

5. Bedingungen (falls zutreffend)

5.1. zur Durchführung der Vorlesung	• Videoprojektor und einer Hochgeschwindigkeits-Internetverbindung
5.2. zur Durchführung des Seminars / der Übung	• Laboraktivität erfordert Computer mit einer hohen Verarbeitungsgeschwindigkeit, einem minimalen Arbeitsspeicher von 16 GB und einer Hochgeschwindigkeits-Internetverbindung.

6.1. Spezifische erworbene Kompetenzen¹

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 K 6.1 Identifizierung der Konzepte und Modelle für Rechnersysteme und Rechnernetze K 6.2 Identifizierung und Erklärung der Basisarchitektur für die Verwaltung vernetzter Rechnersysteme K6.3 Anwendung der Methoden für die Installation, Konfiguration und Verwaltung von Rechnersystemen und Rechnernetzen K6.4 Leistungsmessungen der Antwortzeiten, Ressourcenverbrauch, Festlegen der Zugriffsrechte K6.5 Entwurf von Rechnernetzen
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

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.

Kennt-nisse	- Der Student verfügt über die notwendigen Kenntnisse für die Nutzung von Computern, die Entwicklung von Programmen und Softwareanwendungen, die Informationsverarbeitung. - Der Student verfügt über Kenntnisse in den Bereichen Programmierung, Mathematik, Ingenieurwesen und Technologie und besitzt die notwendigen Fähigkeiten, um diese bei der Erstellung komplexer Computersysteme einzusetzen. - Der/die Studierende verfügt über die erforderlichen Kenntnisse zur Konzeption, Analyse und Verwaltung von Datenbanken. - Der Student verfügt über angemessene Kenntnisse in Bezug auf die Verwendung integrierter Entwicklungsumgebungen für die Erstellung großer komplexer Anwendungen. - Der Student verfügt über Kenntnisse in mehreren Programmiersprachen und ist in der Lage, Anwendungen in kompilierten, interpretierten oder dynamischen Sprachen zu schreiben, wobei er in der Lage ist, die geeignete Programmiersprache für die spezifische zu entwickelnde Anwendung auszuwählen. - Der Student ist in der Lage, geeignete Trainingsverfahren auszuwählen und anzuwenden, um die Aneignung von Wissen zu erleichtern. - Der Student verfügt über die notwendigen Kenntnisse, um die Fachliteratur zu sichten. - Der Student verfügt über die notwendigen Kenntnisse, um Daten und Informationen zu verarbeiten und zu verifizieren. - Der/die Studierende verfügt über die notwendigen Kenntnisse in Bezug auf die Phasen des Software-Lebenszyklus und die Software-Prozessmodelle. - Der Studierende kennt die Konzepte der Softwaremodellierung und ist in der Lage, funktionale und nicht-funktionale Anforderungen, die in spezifischen Dokumenten für die Analyse und den Entwurf von Softwaresystemen beschrieben sind, umzusetzen. - Die Studierenden kennen die Konzepte der Softwaremodellierung und sind in der Lage, die in spezifischen Dokumenten beschriebenen funktionalen und nicht-funktionalen Anforderungen für die Analyse und den Entwurf von Softwaresystemen umzusetzen. - Die Studierenden verfügen über das Wissen, um modellbasierte Softwareentwicklungstechniken anzuwenden. - Der Student verfügt über die notwendigen Kenntnisse im Umgang mit der Sprache UML sowie über die Fähigkeit, CASE-Tools zum Verstehen, Dokumentieren und Implementieren von Softwaresystemen einzusetzen. - Der Studierende kennt die grundlegenden Aspekte des Softwaremanagements. - Der Student ist vertraut mit traditionellen und agilen Entwicklungsmethoden. - Die Studierenden kennen Methoden zum Testen und Verifizieren von Softwaresystemen. - Der/die Studierende kennt die Grundlagen der betriebssystemspezifischen Programmierung und verfügt über grundlegende Kenntnisse der Programmierung in Skriptsprachen. - Der/die Studierende hat grundlegende Kenntnisse in der Installation, Konfiguration und Wartung eines Serversystems im Internet. - Der Student verfügt über angemessene Kenntnisse der Protokolle, mit denen das Internet arbeitet, und besitzt die notwendigen Fähigkeiten, um eigene Protokolle zu entwerfen und zu testen. - Der Student verfügt über die notwendigen Kenntnisse der Internetsicherheit und ist in der Lage, diese Kenntnisse anzuwenden, um ein Computernetz zu validieren und zu warten, das gemeinsame Dienste anbietet, die von außen sicher zugänglich sind. - Der Student ist vertraut mit Werkzeugen, die zum Testen, Debuggen und Validieren von Softwareanwendungen verwendet werden. - Der/die Studierende ist vertraut mit Projektmanagement-Tools, Versionskontrollsystemen und Konzepten, Methoden und Tools der kontinuierlichen Integration/kontinuierlichen Bereitstellung
---------------------------	--

Fähigkeiten	- Der Student verfügt über die notwendigen Fähigkeiten, um Computerprogramme zu entwerfen und Softwaresysteme zu analysieren. - Der/die Studierende ist in der Lage, verschiedene Informationen zu kombinieren, um Lösungen zu formulieren und Entwicklungsideen für neue Produkte und Anwendungen zu entwickeln. - Die Studierenden sind in der Lage, Architektur- und Designvorlagen sowie Best Practices der Industrie anzuwenden, um Softwareanwendungen von hoher Komplexität zu entwerfen. - Der Student ist in der Lage, verschiedene mögliche Architekturen und Lösungen für ein Problem zu bewerten und die geeignete für die spezifischen Anforderungen und Beschränkungen der zu entwickelnden Anwendung auszuwählen. - Der/die Studierende verfügt über die notwendigen Fähigkeiten, um die Konzepte der objektorientierten Programmierung zu verstehen und anzuwenden, um Softwareanwendungen mittlerer Komplexität zu entwickeln. - Der Student ist in der Lage, Designvorlagen für die Anwendungsentwicklung zu verstehen und zu verwenden. - Der Student verfügt über die notwendigen Fähigkeiten, um Anwendungen mit grafischen Benutzeroberflächen unter Verwendung von Architekturvorlagen zu entwickeln, die den Besonderheiten von Anwendungen mit Benutzerinteraktion entsprechen. - Der Student ist in der Lage, vorhandene Module und Umgebungen für die Anwendungsentwicklung auszuwählen und zu nutzen. - Der Studierende ist in der Lage, automatisierte Tests unterschiedlicher Granularität zur Qualitätssicherung der entwickelten Systeme zu erstellen. - Der Student ist in der Lage, didaktisch-pädagogische Ansätze, die auf die kognitive Entwicklung des Schülers abzielen, unter Verwendung spezifischer Strategien und Methoden des Computerunterrichts umzusetzen. - Der/die Studierende hat die Fähigkeit, Informationen zu verstehen und effektiv zu kommunizieren. - Der Student ist in der Lage, Methoden, Algorithmen, Paradigmen und Techniken, die in verschiedenen Zweigen der Informatik verwendet werden, darzustellen und zu erklären. - Der/die Studierende ist in der Lage, Forschungsprobleme in der Informatik zu definieren/identifizieren/verstehen. - Der/die Studierende ist mit internationalen akademischen Forschungsdatenbanken und digitalen Bibliotheken (Web of Science, ACM Digital Library, IEEE Xplore, Springer, Elsevier, CiteSeerX, etc.) vertraut. - Der/die Studierende hat die Fähigkeit, verschiedene Methoden und Werkzeuge zur Visualisierung von Forschungsergebnissen anzuwenden. - Der/die Studierende hat die Fähigkeit, Informationen aus verschiedenen Quellen zu beobachten und zu beschaffen.
Verantwortung und Autonomie	- Die Studierenden sind in der Lage, Programmierparadigmen (prozedural, objektorientiert, funktional) für die Realisierung von Softwareanwendungen auszuwählen und zu verwenden, die für den spezifischen Bereich der entwickelten Anwendung geeignet sind. - Der/die Studierende ist in der Lage, neue Anwendungen, Systeme oder Produkte zu entwickeln, zu entwerfen und zu erstellen, wobei er/sie die besten Praktiken in diesem Bereich anwendet. - Der/die Studierende ist in der Lage, komplexe Probleme zu erkennen und damit verbundene Probleme zu untersuchen, um Lösungsmöglichkeiten zu entwickeln und Lösungen zu implementieren. - Der/die Studierende ist in der Lage, allgemeine Regeln auf spezifische Probleme anzuwenden und entsprechende Lösungen zu erarbeiten. - Der Student hat die Fähigkeit, Werkzeuge zur Unterstützung der Forschung zu nutzen.

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	• Entwickeln der Fähigkeit zu analysieren, zu entwerfen und zu implementieren mobile Geräte Anwendungen
7.2 Spezifische Ziele der Lehrveranstaltung	• Vertraut machen mit der Android Studio Entwicklungsumgebung für mobile Geräte • Fähigkeit Mobile Programme zu entwickeln

8. Inhalt

8.1 Vorlesung	Lehr-und Lernmethode	Anmerkungen
1. Einführung, Betriebssysteme und Technologie für mobile Geräte	Interaktive Vorstellung Konversation	2 Stunden

2. Android-Einführung und Grundelemente I	Ausführung Problematisierung Lehrdemonstration	2 Stunden
3. Android-Einführung und Grundelemente II		2 Stunden
4. Android - grafische Benutzeroberfläche		2 Stunden
5. SQLite - Lokale Datenbank		2 Stunden
6. Verteilte Anwendungen		2 Stunden
7. Ortsbasierte Anwendungen		2 Stunden
8. Sensor basierte Anwendungen (I)		2 Stunden
9. Sensor basierte Anwendungen (II)		2 Stunden
10. Kontextbasierte Anwendungen		2 Stunden
11. Cross-Platform Anwendungen (React Native, Flutter)		2 Stunden
12. Einführung in iOS und Windows Phone		2 Stunden
13. Erweiterte Funktionen zum Erstellen komplexer mobiler Anwendungen		2 Stunden
14. Präsentation und Diskussion von Projekten	Interaktive Vorstellung Konversation	2 Stunden
Literatur in deutscher Sprache 1. Hauke Fehr, "Eigene Apps programmieren: Schritt für Schritt mit LiveCode zur eigenen App – für Windows, Mac, iOS und Android", Editor Rheinwerk Verlag GmbH, 2016, ISBN-13: 978-3836243803 2. Vanda Nahavandipour, „React Native Native Apps parallel für Android und iOS entwickeln“, ISBN Print: 978-3-96009-066-3 3. Jörg Staudemeyer, „Android mit Kotlin – kurz & gut: Inklusive Android 8 und Android Studio 3.0“, O'Reilly 2018, ISBN-13: 978-3960090380. 4. Becker, A., & Pant-Android, M. (2009). Grundlagen und Programmierung. dpunkt Verlag. 5. Becker, Arno, and Marcus Pant. "Android 2." Heidelberg: dpunkt (2010). 6. Thomas Theis, „Einstieg in Kotlin: Apps entwickeln mit Android Studio“, Rheinwerk Verlag, 2. aktualisierte Auflage, 2021. 7. Marc Marburger, „Flutter und Dart: Das umfassende Handbuch für die professionelle App-Entwicklung“, Rheinwerk Verlag, 1. Auflage, 2021.		
Literatur in andere Sprachen 8. Android Development. http://developer.android.com/index.html 9. Vogella. Android Development Tutorials. http://www.vogella.com/android.html 10. Michael Y. Morckos, Android Architecture, German University in Cairo, May 13, 2009 11. Reto Meier, Professional Android 2 Application Development, 2010 12. M. Cremene, I. K. Bența, „Dezvoltarea de aplicatii pentru terminale mobile“, Ed. Alabastra, Cluj- Napoca, 2006.		
8.2 Seminar / Laborarbeit	Lehr-und Lernmethode	Anmerkungen
1. Präsentation von Projektthemen, Analyse und Design von Anwendungen	Konversation, Entdeckung, Selbststudium, Brainstorming	2 Stunden
2. Einfache Anwendungen in Android Studio, Entwerfen und Erstellen der grafischen Benutzeroberfläche	Konversation, Algorithmisierung, Problematisierung, Zusammenarbeit, Selbststudium, Übung	2 Stunden
3. Lokale Datenbanken Design und Implementierung		2 Stunden
4. Verteilte Anwendungen		2 Stunden
5. Orts und Sensor basierte Anwendungen		2 Stunden
6. Entwickeln von Anwendungen aus dem Projekt	Konversation, Algorithmisierung, Fallstudie, Simulation, Zusammenarbeit, Selbststudium	2 Stunden
7. Verfeinerung und Vervollständigung von Anwendungen aus dem Projekt	Konversation, Problematisierung, Algorithmisierung, Fallstudie	2 Stunden
Literatur in deutscher Sprache 1. Hauke Fehr, "Eigene Apps programmieren: Schritt für Schritt mit LiveCode zur eigenen App – für Windows, Mac, iOS und Android", Editor Rheinwerk Verlag GmbH, 2016, ISBN-13: 978-3836243803		

2. Vanda Nahavandipour, „React Native Native Apps parallel für Android und iOS entwickeln“, ISBN Print: 978-3-96009-066-3
 3. Jörg Staudemeyer, „Android mit Kotlin – kurz & gut: Inklusive Android 8 und Android Studio 3.0“, O'Reilly 2018, ISBN-13: 978-3960090380.
 4. Becker, A., & Pant-Android, M. (2009). Grundlagen und Programmierung. dpunkt Verlag.
 5. Becker, Arno, and Marcus Pant. "Android 2." Heidelberg: dpunkt (2010).
 6. Thomas Theis, „Einstieg in Kotlin: Apps entwickeln mit Android Studio“, Rheinwerk Verlag, 2. aktualisierte Auflage, 2021.
 7. Marc Marburger, „Flutter und Dart: Das umfassende Handbuch für die professionelle App-Entwicklung“, Rheinwerk Verlag, 1. Auflage, 2021.
- Literatur in andere Sprachen
8. Android Development. <http://developer.android.com/index.html>
 9. Vogella. Android Development Tutorials. <http://www.vogella.com/android.html>
 10. Michael Y. Morckos, Android Architecture, German University in Cairo, May 13, 2009
 11. Reto Meier, Professional Android 2 Application Development, 2010
 12. M. Cremene, I. K. Bența, „Dezvoltarea de aplicatii pentru terminale mobile“, Ed. Alabastra, Cluj- Napoca, 2006.

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

- Der Kurs respektiert curriculare IEEE und ACM Empfehlungen für Studium der Informatik
- Der Kurs existiert im Lehrplan der meisten Fakultäten in Rumänien
- Der Kurs existiert in viele Fakultäten auf der ganzen Welt
- Softwareunternehmen betrachten den Inhalt des Kurses als nützlich, um die Modellierungs- und Programmierfähigkeiten der Schüler zu entwickeln

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	- Kenntnis der grundlegenden Konzepte des Feldes - Anwendung von OOPPrinzipien und mobilspezifischen Technologien zur Entwicklung realer Anwendungen	Schriftliche Prüfung	30%
10.5 Seminar / Übung	- Analyse, Design, Implementierung und Test von mobilen Anwendungen aus dem Projekt	Projekt Präsentation	50%
	- Analyse, Design, Implementierung und Test von mobilen Anwendungen aus dem Labor	Systematische Beobachtung des Studenten bei der Lösung der Laboraufgaben	20%

10.6 Minimale Leistungsstandards

- Jeder Student muss nachweisen, dass er / sie ein akzeptables Wissen und Verständnis auf diesem Gebiet erreicht hat, dass er in der Lage ist, Wissen in einer kohärenten Form auszudrücken, dass er in der Lage ist, bestimmte Verbindungen herzustellen und Wissen zu nutzen Design und Implementierung von Anwendungen, die echte Probleme lösen
- Um die Prüfung zu fördern, muss man:
 - um die Prüfung zu fördern, muss man ein originelles Minimal mit einer grafischen Interface-Applikation und einer zusätzlichen Einrichtung machen (mit Hilfe einer Datenbank, Sensoren

oder GPS)

- der Durchschnitt der Bewertung (schriftliche Prüfung, Labor) liegt über der Note 5

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

Nicht anwendbar.

Ausgefüllt am:
April 2025

Vorlesungsverantwortlicher
Lektor dr. ing. Kuderna-Iulian Bența

Seminarverantwortlicher
Lektor dr. ing. Kuderna-Iulian Bența

Genehmigt im Department am:
April 2025

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*“.