

LEHRVERANSTALTUNGSBESCHREIBUNG

Logische und Funktionale Programmierung

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	Logische Programmierung					Code der LV	MLG5143
2.2. Lehrverantwortlicher – Vorlesung		Conf. Dr. Christian Sacarea					
2.3. Lehrverantwortlicher – Seminar		Asist. Dr. Florin Albişoru					
2.4. Studienjahr		2.5. Semester		2.6. Prüfungsform	C	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	1+1
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					25
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					20
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays					30
Tutoriat (consiliere profesională)					10
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	Grundbegriffe der Programmierung, Logik, Datenstrukturen und Algorithmen
4.2. kompetenzbezogene	-

5. Bedingungen (falls zutreffend)

5.1. zur Durchführung der Vorlesung	· Videoprojektor
5.2. zur Durchführung des Seminars / der Übung	· GCLisp, CLisp, TurboProlog

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 • K1.5 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 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	• die Vertrautheit mit deklarativen Programmierung
--	--

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	• die Einführung einer Programmiersprache für jedes Paradigma (CLisp, Prolog) • die Idee der Verwendung dieser Paradigmen für die Bedürfnisse der Softwareprogramme • die Grundlagen für nachfolgende fortgeschrittene Programmierkurse anzubieten
--	--

8. Inhalt

8.1 Vorlesung	Lehr-und Lernmethode	Anmerkungen
Logische Programmierung. PROLOG		
1. Rekursion und Programmiersprachen. Imperative Programmierung und deklarative Programmierung. Einführung. Rekursion. Beispiele.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
2. Grundlagen der Prolog-Programmierung. Prolog Fakten und Regeln. Fragen. Kontrollstrukturen in Prolog. Variable und zusammengesetzte Terme. Anonyme Variable. Planungssysteme. Die Teile eines Prolog-programms. Beispiele.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
3. Das Prolog-programm. Vordefinierte Domäne. Interne und externe Fragen. Multiplizität der Prädikate. Das IF Symbol und der IF Befehl. Arithmetik und Vergleiche. Eingabe und Ausgabe . Zeichenketten	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
4. Backtracking. Die "Fail" und "!" Prädikate. Das "Not" Prädikat. Listen. Rekursion. Beispiele.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
5. Zusammengesetzteobjekte und Funktoren (bzw "Funktionsobjekte"). Die Vereinigung der Zusammengesetzte Objekte. Vergleich von Zusammengesetzte Objekte. Backtracking mit Zyklen. Beispiele der rekursiven Programmierung. Der StackFrame. Tail Recursion und Optimierung.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
6. Rekursive Datenstrukturen. Bäume als Datenstruktur. Suche in Bäumen. Suchbäume. Die intern Datenbank der Prolog.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
7. Prüfung		
Funktionale Programmierung. LIS		
8. Die Bedeutung der funktionalen Programmierung als ein neues Programmierparadigma. Einführung in Lisp. Grundelemente von Lisp. Dynamische Datenstrukturen. Semantische und syntaktische Regeln. Klassifikation der Lisp-Funktionen. Primitive Funktionen.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
9. Lisp Prädikate. Logische und arithmetische Funktionen. Benutzerdefinierte Funktionen. Beispiele.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
10. Die Verwaltung der Symbole. Weitere Listenbefehle. OBLIST, ALIST. Weitere interessante Funktionen. Beispiele.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
11. Die Eval-Funktion. Die FUNCALL und APPLY Funktionen. Lambda-Funktionen. Label-Funktionen. Beispiele.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	

12. Generators. MAP-Funktionen. Beispiele.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
13. Weitere Grundelemente.	Beschreibung, Erklärungen, Unterrichtsgespräch, Vorführung	
14. Prüfung		
Literatur in deutscher Sprache - Goos, G., Zimmermann, W., Vorlesungen über Informatik, Band 1, Grundlagen und funktionales Programmieren, Springer, 2006. - Lippe, W-M., Funktionale und Applikative Programmierung, Springer 2009. - Cordes/Kruse/Langendorfer/Rust: Prolog — Eine methodische Einführung Vieweg, 1990, 2. Auflage, 245 Seiten - Hanus: Problemlosen mit Prolog Teubner, 1987, 2. Auflage, 224 Seiten - Kleine Buning/Schmitgen: Prolog, Teubner, 1988, 2. Auflage, 311 Seiten Allgemeine Literatur: - Czibula, G., Pop, H.F., Elemente avansate de programare în Lisp și Prolog. Aplicații în Inteligența Artificială, Editura Albastră, Cluj-Napoca, 2012 - Pop, H.F., Șerban, G., Programare în Inteligența Artificială - Lisp și Prolog, Editura Albastră, Cluj-Napoca, 2003 http://www.lpa.co.uk, Logic Programming - Field, A., Functional Programming, Addison Wesley, New York, 1988. - Winston, P.H., Lisp, Addison Wesley, New York, 2nd edition, 1984.		
8.2 Seminar / Übung	Lehr-und Lernmethode	Anmerkungen
1. Rekursion	Erklärungen, Unterrichtsgespräch, Modellierung	
2. Listen	Erklärungen, Unterrichtsgespräch, Modellierung	
3. Bäume	Erklärungen, Unterrichtsgespräch, Modellierung	
4. Backtracking	Erklärungen, Unterrichtsgespräch, Modellierung	
4. Praktische Klausur		1 Stunde
5. Rekursive Programmierung	Erklärungen, Unterrichtsgespräch, Modellierung	
6. MAP-Funktionen	Erklärungen, Unterrichtsgespräch, Modellierung	
7. Iterative Programmierung	Erklärungen, Unterrichtsgespräch, Modellierung	
7. Praktische Klausur		1 Stunde
Literatur in deutscher Sprache - Hölldobler, S., Logik und Logikprogrammierung: Synchron Wissenschaftsverlag der Autoren, Heidelberg, 2001 - Weisweber, W., Logische Programmierung in der Praxis, Redline GmbH, 2000 - Lippe, W-M., Funktionale und Applikative Programmierung, Springer 2009. Allgemeine Literatur		

- Czibula, G., Pop, H.F., Elemente avansate de programare în Lisp și Prolog. Aplicații în Inteligența Artificială, Editura Albastră, Cluj-Napoca, 2012
- Documentatia produselor: Gold Common Lisp 1.01 si 4.30, XLisp, Free Lisp.

<http://www.swi-prolog.org/>

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

- Der Kurs existiert in der Mehrzahl der rumänischen und ausländischen Universitäten.
- Der Kursinhalt bietet die notwendigen Lisp und Prolog Programmierkenntnisse für eine eventuelle Arbeitsstelle.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	Die Richtigkeit und die Vollständigkeit der erworbenen Kenntnisse	Schriftliche Prolog-Klausur (während des Semesters)	30%
		Schriftliche Lisp-Klausur (während des Semesters)	30%
10.5 Seminar / Übung	- Die Umsetzung der Aufgaben in Lisp und Prolog - Die Ausarbeitung der Dokumentation - Die Einhaltung der Frist für die Aufgaben	die Softwaredokumentation	10%
		Praktische Prolog-Klausur (1 Stunde)	15%
		Praktische Prolog-Klausur (1 Stunde)	15%
10.6 Minimale Leistungsstandards			
• Kenntnisse der Grundlagen. Die Lösung einer einfachen Aufgabe in einer deklarativen Programmiersprache. Jeder Student muss einen akzeptablen Kenntnisstand beweisen • Min 5 auf jeder Klausur. Min 5 auf jeder Übung. 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
--	---

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“.

Ausgefüllt am:
11.04.2025

Vorlesungsverantwortlicher
Conf. Dr. Christian Sacarea

Seminarverantwortlicher
Lect. dr. Florin Albisoru

Genehmigt im Department am:
25.04.2025

Departmentleiter
Conf. Dr. Adrian Sterca