

Lehrende/r	Christoph Beierle		Modulbeauftragte/r		Christoph Beierle	
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden	Häufigkeit in jedem Wintersemester		
Lehrveranstaltung(en)	01816	Logisches und funktionales Programmieren			WS	SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 130 - 150 Stunden bearbeiten der Übungsaufgaben: 60 - 75 Stunden Prüfungsvorbereitung: 60 - 75 Stunden					
Qualifikationsziele	Die Studierenden können grundlegende Aspekte nichtimperativer Programmiersprachen und deren wichtigste Programmiertechniken beschreiben. Sie sind in der Lage, diese Techniken in logischen und funktionalen Programmen anzuwenden. Sie können die spezielle Ausdruckskraft und Flexibilität des logischen und funktionalen Programmierens erklären und diese bei der Erstellung logischer und funktionaler Programme demonstrieren.					
Inhalte	Der Kurs enthält eine Einführung in die wichtigsten Programmiertechniken für nicht-imperative Programmiersprachen. Im ersten Teil werden Techniken der logischen Programmierung am Beispiel von Prolog gezeigt. Außerdem werden die in Prolog verwendete Beweisstrategie sowie nicht-logische Anteile zur praktischen Programmierung behandelt. Im zweiten Teil werden Techniken aus funktionalen Programmiersprachen am Beispiel des Lisp-Dialekts Scheme demonstriert. Neben klassischen Methoden (Funktionen höherer Ordnung) werden auch Datenabstraktion, strom- und objektorientierte Programmiertechniken vorgestellt. Im dritten Teil werden Ansätze zur Integration der logischen und funktionalen Programmierung vorgestellt. Darüber hinaus werden verschiedene neuere Entwicklungen im Bereich dieser beiden Programmierparadigmen behandelt. So wird auch auf das logische Programmieren mit Constraints eingegangen, das sich als besonders vielversprechender Ansatz zur deklarativen Programmierung herausgestellt hat und bereits in vielen Anwendungen erfolgreich eingesetzt wurde.					
	Ergänzende Literatur: W. F. Clocksin, C. S. Mellish: Programming in Prolog. Springer, Berlin, 5. Auflage, 2003 L. Sterling, E. Shapiro: The Art of Prolog. MIT Press, 2. Auflage, 1994. H. Abelson, G. J. Sussman: Structure and Interpretation of Computer Programs, MIT Press, 2. Auflage, 1996.					
Inhaltliche Voraussetzung	Vertrautheit mit algorithmischer Denkweise. Kenntnis wenigstens einer höheren Programmiersprache und gängiger Datenstrukturen wie Listen und Bäume.					
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum					
Anmerkung	Das Modul kann letztmalig im WS 2021/22 belegt werden. Eine Prüfungsteilnahme ist nur noch bis einschließlich SS 2022 möglich.					
Formale Voraussetzung	keine					
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik					

Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete Prüfungsklausur

keine

Stellenwert
der Note 1/12