

Modul INF-MSc-326: Logische Methoden des Software Engineering 2 (LMSE2)¹					
Englischer Modultitel: Logic Methods in Software Engineering 2					
Studiengänge: Masterstudiengang Informatik, Masterstudiengang Angewandte Informatik					
Turnus nach Ankündigung	Dauer 1 Semester	Studienabschnitt 2.-3. Semester	Credits 6	Aufwand 180 (60/120)	
1	Modulstruktur				
	Nr.	Element / Lehrveranstaltung	Typ	Credits	SWS²
	1	Logische Methoden des Software Engineering 2	V	3	2
	2	Übung und Praktikumsprojekt zu Logische Methoden des Software Engineering 2	Ü+P	3	2
2	Lehrveranstaltungs-sprache: deutsch				
3	Lehrinhalte Die Vorlesung umfasst die folgenden Lehrinhalte: <ul style="list-style-type: none"> • Intersektionstypen für den Lambda-Kalkül und die kombinatorische Logik mit Anwendungen bei der Synthese und Komposition von Komponenten. • Polymorphismus (System F, polymorpher Lambda-Kalkül der zweiten Ordnung). • Parametrisierte Typen (Typfamilien, dependent types) und Anwendungen in Programmiersprachen, in Theorembeweissystemen und bei Programmkorrektheitsbeweisen. Die begleitenden Übungen zu Logische Methoden des Software Engineering 2 dienen zur Vertiefung des in der Vorlesung behandelten Stoffes. Dies geschieht durch regelmäßig ausgegebene Übungsaufgaben, die die Studierenden selbstständig bearbeiten. In den Präsenzzeiten der Übung werden die Lösungen der Aufgaben in kleineren Übungsgruppen besprochen. Im begleitenden Praktikumsprojekt werden Vorlesungsinhalte selbstständig praktisch zum Lösen von Programmieraufgaben angewandt. Lösungen werden in Kleingruppen erstellt und anschließend von den Studierenden präsentiert.				
4	Kompetenzen Die Studierenden sollen ausgewählte mächtigere Typsysteme studieren und dadurch die Struktur der weitergehenden Typentheorie kennen lernen, sowie ausgewählte Anwendungen in Programmiersprachen, in Theorembeweissystemen, bei Programmkorrektheitsbeweisen und bei Programmsynthese. Die Studierenden sollen zentrale Resultate der Metatheorien kennen, einschließlich Unentscheidbarkeitseigenschaften. Sie sollen in die Lage versetzt werden, die grundlegende Architektur von Theorembeweissystemen zu verstehen und diese bei einfachen Beweiskonstruktionsproblemen einzusetzen. Darüber hinaus sollen die Studierenden die Architektur von weitergehenden Typsystemen für moderne funktionale Programmiersprachen verstehen, insbesondere die Verwendung von parametrisierten Typen. Außerdem sollen die Studierenden Intersektionstypen verstehen und Anwendungen des Inhabitationsproblems bei Programmsyntheseproblemen verstehen.				
5	Prüfungen <i>Modulprüfung:</i> Klausur (max. 60 Minuten) ^{BOSS-NR. ?????} <i>Studienleistung:</i> <ul style="list-style-type: none"> • Übungs- und Praktikums-schein in Element 2 gemäß Ankündigung des Prüfers oder der Prüferin ^{BOSS-NR. ?????} Die Studienleistung ist Voraussetzung für die Teilnahme an der Modulprüfung.				
6	Prüfungsformen und -leistungen <input checked="" type="checkbox"/> Modulprüfung <input type="checkbox"/> Teilleistungen				
7	Teilnahmevoraussetzungen				

¹ Bei Wahl dieses Moduls ist die Wahl des Moduls INF-MSc-312 „Komponenten- und Service-Orientierte Softwarekonstruktion“ ausgeschlossen.

² Dieses Modul wird in der Regel in der zweiten Hälfte der Vorlesungszeit mit wöchentlich vier Stunden Vorlesung und vier Stunden Übungen/Praktikum angeboten. Das Modul LMSE1, dessen Inhalte für das Absolvieren dieses Moduls vorausgesetzt werden, wird in der Regel in der ersten Hälfte der Vorlesungszeit desselben Semesters angeboten.

	<i>Vorausgesetzte Kenntnisse:</i> Modul „Logische Methoden des Software Engineering 1“ <i>Wünschenswerte Kenntnisse:</i> Programmiererfahrung in einer funktionalen Programmiersprache		
8	Modultyp und Verwendbarkeit des Moduls Vertiefungsmodul in den Masterstudiengängen Informatik und Angewandte Informatik Forschungsbereich: Software, Sicherheit und Verifikation		
9	Modulbeauftragte/r Prof. Dr. J. Rehof	Zuständige Fakultät Informatik	Beschluss Fakultätsrat 17.10.2018