

Modul INF-MSc-325: Logische Methoden des Software Engineering 1 (LMSE1¹)					
Englischer Modultitel: Logic Methods in Software Engineering 1					
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 1	V	3	2
	2	Übung und Praktikumsprojekt zu Logische Methoden des Software Engineering 1	Ü+P	3	2
2	Lehrveranstaltungssprache: deutsch				
3	<p>Lehrinhalte</p> <p>Die Vorlesung umfasst die folgenden Lehrinhalte:</p> <ul style="list-style-type: none"> • Einführung in den ungetypten Lambda-Kalkül, u.a. beta-Reduktion, Church-Rosser Satz und Turing-Vollständigkeit. • Der einfach getypte Lambda-Kalkül und dessen Metatheorie (u.a. Subject Reduction, schwache und starke Normalisierung). • Kombinatorische Logik. • Curry-Howard Isomorphismus und konstruktive Logik. • Das Typisierbarkeitsproblem und das Inhabitationsproblem. <p>Die begleitenden Übungen zu Logische Methoden des Software Engineering 1 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.</p> <p>Im begleitenden Praktikumsprojekt werden Vorlesungsinhalte selbstständig praktisch zum Lösen von Programmieraufgaben angewandt. Lösungen werden in Kleingruppen erstellt und anschließend von den Studierenden präsentiert.</p>				
4	<p>Kompetenzen</p> <p>Die Studierenden lernen die Grundlagen der Typentheorie und der konstruktiven Logik kennen. Sie sollen die grundlegende Theorie des ungetypten Lambda-Kalküls sowie die Metatheorie des einfach getypten Lambda-Kalküls beherrschen. Sie sollen in der Lage sein, zentrale Sätze dieser Theorien zu beweisen und sie sollen Verständnis für deren Bedeutung bei Anwendungen in Programmiersprachen erwerben. Sie sollen die algorithmische Theorie von wesentlichen Entscheidungsproblemen (insb. Typisierbarkeit und Inhabitation) in dem Umfeld verstehen.</p> <p>Die Studierenden werden durch das Praktikumsprojekt in die Lage versetzt, Veranstaltungsinhalte selbstständig praktisch anzuwenden.</p>				
5	<p>Prüfungen</p> <p><i>Modulprüfung:</i> Klausur (max. 60 Minuten) <small>BOSS-NR. ?????</small></p> <p><i>Studienleistung:</i></p> <ul style="list-style-type: none"> • Übungs- und Praktikumschein in Element 2 gemäß Ankündigung des Prüfers oder der Prüferin <small>BOSS-NR. ????</small> <p>Die Studienleistung ist Voraussetzung für die Teilnahme an der Modulprüfung.</p>				
6	<p>Prüfungsformen und -leistungen</p> <p><input checked="" type="checkbox"/> Modulprüfung <input type="checkbox"/> Teilleistungen</p>				

¹ Bei Wahl dieses Moduls ist die Wahl des Moduls INF-MSc-319 „Logische Methoden des Software Engineering“ ausgeschlossen.

² Dieses Modul wird in der Regel in der ersten Hälfte der Vorlesungszeit mit wöchentlich vier Stunden Vorlesung und vier Stunden Übungen/Praktikum angeboten. Das Modul LMSE2 wird in der Regel in der zweiten Hälfte der Vorlesungszeit desselben Semesters angeboten.

7	Teilnahmevoraussetzungen <i>Vorausgesetzte Kenntnisse:</i> Verständnis für Grundbegriffe der theoretischen Informatik, insbesondere der Berechenbarkeits- und Komplexitätstheorie <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