

| | | | | | |
|--|--|--|--|--------------------------------|---|
| Modul INF-MSc-304: Funktionales und regelbasiertes Programmieren | | | | | |
| Englischer Modultitel: Functional and Rule-based Programming | | | | | |
| 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 | SWS |
| | 1 | Funktionales und regelbasiertes Programmieren | V | 3 | 2 |
| | 2 | Übung zu Funktionales und regelbasiertes Programmieren | Ü | 3 | 2 |
| 2 | Lehrveranstaltungssprache: deutsch oder englisch | | | | |
| 3 | Lehrinhalte Die Lehrveranstaltung behandelt Konzepte und Konstrukte funktionaler und regelbasierter Programmierung und deren Realisierung in den Programmiersprachen Haskell, O'Haskell und Maude. Neben ihrem Einsatz in den üblichen Anwendungsbereichen funktionaler Programmierung werden mit O'Haskell zustandsbasierte reaktive Systeme wie z.B. GUIs auf einer problemnahen Ebene implementiert, während sich mit Maude insbesondere nichtdeterministische Transitionssysteme, die auf strukturierten Zuständen mit verteilten Komponenten (z.B. high-level Petri-Netze) einfach und effizient realisieren lassen. Zur Illustration der Konzepte in einem nichttrivialen Rahmen wie auch zum Test und zur Verifikation einzelner Programme wird das rapid-prototyping tool Expander2 benutzt. | | | | |
| 4 | Kompetenzen Die Studierenden üben die Verwendung von Sprachen, die nicht nur zur Implementierung, sondern bereits zum Entwurf auf höheren Abstraktionsebenen geeignet sind. Die Benutzung solcher Sprachen fördert die Fähigkeit, Software auf einer soliden Grundlage präziser mathematischer Konzepte zu entwickeln und gleichzeitig die Ausführbarkeit im Auge zu haben, weil eben beides mit derselben Sprache geschehen kann. Diese Kompetenz ist heute kein Luxus mehr, sondern notwendig, um in der Fülle angebotener Entwurfs- und Implementierungswerkzeuge die für den jeweiligen Anwendungsbereich und die jeweiligen Constraints geeigneten auszuwählen zu können. | | | | |
| 5 | Prüfungen <i>Modulprüfung:</i> mündliche Prüfung (30 Minuten) ^{BOSS-NR. 63491} <i>Studienleistung:</i> • Aktive Teilnahme (inkl. Präsentation eigener Lösungen) Erreichen einer Mindestpunktzahl der Übungsaufgaben ^{BOSS-NR. 63441} 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 <i>Erfolgreich abgeschlossen:</i> Ein Basismodul aus dem Forschungsbereich Software, Sicherheit und Verifikation <i>Vorausgesetzte Kenntnisse:</i> Mindestens eine funktionale oder eine objektorientierte Programmiersprache, Begriffe, Methoden und Informatik-Anwendungen von diskreter Mathematik, Algebra und Logik | | | | |
| 8 | Modultyp und Verwendbarkeit des Moduls Vertiefungsmodul in den Masterstudiengängen Informatik und Angewandte Informatik Forschungsbereich: Software, Sicherheit und Verifikation | | | | |
| 9 | Modulbeauftragte/r Prof. Dr. P. Padawitz | | Zuständige Fakultät Informatik | | Beschluss Fakultätsrat 13.01.2010 Änderung Fakultätsrat 21.09.2016 |