

INF-BSc-213: Funktionale Programmierung (FuPro)					BOSS-Nr. 61800	
Englischer Modultitel: Functional Programming						
Studiengänge: Bachelorstudiengang Informatik , Bachelorstudiengang Angewandte Informatik						
Turnus: jährlich im Sommersemester		Dauer: 1 Semester	Studienabschnitt: Ab 5. Semester		Credits: 4	Aufwand: 120 (45/75)
1	Modulstruktur					
	Nr.	Element / Lehrveranstaltung		Typ	Credits	SWS
	1	Funktionale Programmierung		Vorlesung	3	2
	2	Übungen zu Funktionale Programmierung		Übung	1	1
2	Lehrveranstaltungs-sprache: deutsch					
3	<p>Lehrinhalte</p> <p>Die Lehrveranstaltung führt ein in Konzepte und Anwendungen funktionaler, musterbasierter und monadischer Programmierung anhand der funktionalen Programmiersprache Haskell. Der Kern eines funktionalen Programms ist ein System rekursiver Gleichungen zwischen funktionalen Ausdrücken. Auch die Eingabe- und Ausgabedaten sind funktionale Ausdrücke, wobei hier die Funktionen auf Konstruktoren beschränkt sind, die den Aufbau der Daten beschreiben. Dieses Sprachmodell und der damit einhergehende Programmierstil unterscheiden sich von denen einer objektorientierten, imperativen oder zustandsorientierten Sprache wie Java. Häufig erlauben sie flexible und generische Lösungen, die leicht an neue Anforderungen oder modifizierte Datenstrukturen anpassbar sind. Darüber hinaus lassen sich diese Charakteristika funktionaler Programme mithilfe von Typklassen, insbesondere den monadischen, auch auf zustandsorientierte Lösungen übertragen, die deterministische, nichtdeterministische oder partielle Berechnungen realisieren. Die Ausführung jedes Haskell-Programms folgt der lazy evaluation genannten Strategie, die das Rechnen mit potenziell unendlichen Datenströmen erlaubt. Insgesamt ist Haskell als kompakte Modellierungs- und Entwurfssprache einsetzbar, in der sich formale Modelle direkt ausführen lassen. Neben dem Kennenlernen eines weiteren Programmierstils geht es bei dieser Lehrveranstaltung um die Klassifikation wichtiger Programm- und Datenstrukturen auf der Basis mathematischer Modelle und deren Implementierung in Haskell. Ergänzend zu der Spracheinführung in Haskell betrachtet die Lehrveranstaltung grundlegende theoretische Konzepte von funktionalen Programmiersprachen. Dazu werden unter anderem ausgewählte Themen des Lambda-Kalküls sowie der Kategorien- und Typtheorie betrachtet.</p>					
4	<p>Kompetenzen</p> <p>Nach erfolgreichem Abschluss des Moduls können die Studierenden:</p> <ul style="list-style-type: none"> • Grundkonzepte funktionaler, musterbasierter und monadischer Programmierung mit Haskell erklären und anwenden, • Datenstrukturen in Haskell mithilfe mathematischer Modelle darstellen, • Programme entwerfen, die Konzepte zustandsorientierter Implementierungssprachen umsetzen, • Programme entwerfen, die unendliche Datenströme verarbeiten, • Konzepte des Lambda-Kalküls sowie der Kategorien- und Typtheorie anhand funktionaler Programmierung erklären. 					
5	<p>Prüfungen</p> <ul style="list-style-type: none"> • Modulprüfung: Klausur oder mündliche Prüfung BOSS-NR. 61891 • Studienleistung: BOSS-NR. 61841 • Aktive Mitarbeit in der Übung und Erreichen der Mindestpunktzahl der Übungsaufgaben oder erfolgreiche Teilnahme an einem midterm-Test nach Ankündigung des Veranstalters. 					
6	<p>Prüfungsformen und -leistungen</p> <p>[x] Modulprüfungen [] Teilleistung</p>					

7	Teilnahmevoraussetzungen <ul style="list-style-type: none"> • Erfolgreich abgeschlossen: –keine– • Vorausgesetzte Kenntnisse: Modul „Datenstrukturen Algorithmen und Programmierung 1 (DAP 1)“; „Datenstrukturen Algorithmen und Programmierung 2 (DAP 2)“; Modul „Mathematik für Informatik 1“ oder Modul „Höhere Mathematik 1“, Modul „Mathematik für Informatik 2“ oder Modul „Höhere Mathematik 2“ 		
8	Modultyp und Verwendbarkeit des Moduls <ul style="list-style-type: none"> • Wahlpflicht-Modul im Bachelor-Studiengang Informatik und Angewandte Informatik • Katalog: Konzepte für Software 		
9	Modulbeauftragte/r Prof. Dr. J. Rehof	Zuständige Fakultät: Fakultät für Informatik	Beschluss Fakultätsrat: 03.09.2025