

Praktische Informatik 3 <i>Practical Computer Science 3</i>							Modulnummer: IBGP-PI3		
Bachelor Pflicht/Wahl <input checked="" type="checkbox"/> Wahl <input type="checkbox"/> Basis <input type="checkbox"/> Ergänzung <input type="checkbox"/> Sonderfall <input type="checkbox"/>				Zugeordnet zu Masterprofil Sicherheit und Qualität (SQ) <input type="checkbox"/> KI, Kognition, Robotik (KIKR) <input type="checkbox"/> Digitale Medien und Interaktion (DMI) <input type="checkbox"/>					
Modulbereich: Praktische und Technische Informatik Modulteilbereich: (keine Angabe)									
Anzahl der SWS	V	UE	K	S	Prak.	Proj.	Σ	Kreditpunkte: 6	Turnus angeboten in jedem WiSe
	2	2	0	0	0	0	4		
Formale Voraussetzungen: -									
Inhaltliche Voraussetzungen: Praktische Informatik 2									
Vorgesehenes Semester: 3. Semester									
Sprache: Deutsch									
Ziele: <ul style="list-style-type: none"> • Konzepte und typische Merkmale des funktionalen Programmierens kennen, verstehen und anwenden können. • Vertieftes Verständnis von Datenstrukturen und Algorithmen. • In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können. <p>Die Vorlesung Praktische Informatik 3 vermittelt essenzielles Grundwissen und Basisfähigkeiten, deren Beherrschung für nahezu jede vertiefte Beschäftigung mit Informatik Voraussetzung ist.</p>									
Inhalte: <ol style="list-style-type: none"> 1. Grundlagen der funktionalen Programmierung: Rekursion – Definition von Funktionen durch rekursive Gleichungen und Mustervergleich (pattern matching) – Auswertung, Reduktion, Normalform – Funktionen höherer Ordnung, currying, Typkorrektheit und Typinferenz 2. Typen: Algebraische Datentypen – Typkonstruktoren – Typklassen – Polymorphie – Standarddatentypen (Listen, kartesische Produkte, Lifting) und Standardfunktionen darauf (fold, map, filter) – Listenkomprehension 3. Algorithmen und Datenstrukturen: Unendliche Listen (Ströme) – Bäume – Graphen – zyklische Datenstrukturen 4. Strukturierung und Spezifikation: Module – Schnittstellen (Interfaces) – Abstrakte Datentypen – Signaturen und Axiome 5. Theoretische Aspekte: Referentielle Transparenz – Lambda-Kalkül – Beweis durch Induktion 6. Fortgeschrittene Funktionale Programmierung: Funktionale I/O und zustandsbasierte Programme – Monaden <p>Im Übungsbetrieb; Programmentwicklung in Haskell — Realisierung einzelner, überschaubarer Programmieraufgaben in kleinen Gruppen</p>									
Lehrveranstaltung(en): <ul style="list-style-type: none"> • 03-IBGP-PI3 Praktische Informatik 3: Funktionale Programmierung 									
Unterlagen (Skripte, Literatur, Programme usw.): <ul style="list-style-type: none"> • Simon Thompson: Haskell - The Craft of Functional Programming, Addison-Wesley, 3. Auflage 2011. <p>Weiteres Lehrmaterial ist auf der Webseite des Veranstaltung zu finden:</p> <ul style="list-style-type: none"> • Folienkopien • Übungsaufgaben • Hinweise auf Quellen im WWW <p>Das Haskell-System ghci ist frei verfügbare Software (für Linux, Windows und MacOS).</p>									
Form der Prüfung: KP, PL1: xx%, PL2: xx%, Portfolio, Klausur									

Arbeitsaufwand	Präsenz	56 h
	Übungsbetrieb/Prüfungsvorbereitung	124 h
	Summe	180 h
Lehrende: Prof. Dr. C. Lüth, Dr. Th. Barkowsky		Verantwortlich: Prof. Dr. C. Lüth