

Modulbezeichnung	Praktische Informatik 3								
Modulverantwortliche(r)	Prof. Dr. C. Lüth								
Modulart	Pflicht/Wahl <input checked="" type="checkbox"/> Wahlpflicht <input type="checkbox"/>								
Spezialisierungsbereich									
Dauer des Moduls	1 Semester								
Kreditpunkte	6 CP								
Arbeitsaufwand	<table style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2">Berechnung des Workloads</td> </tr> <tr> <td style="width: 80%;">Präsenz</td> <td style="text-align: right;">56 h</td> </tr> <tr> <td>Übungsbetrieb/Prüfungsvorbereitung</td> <td style="text-align: right;">124 h</td> </tr> <tr> <td style="border-top: 1px solid black;">Summe</td> <td style="text-align: right; border-top: 1px solid black;">180 h</td> </tr> </table>	Berechnung des Workloads		Präsenz	56 h	Übungsbetrieb/Prüfungsvorbereitung	124 h	Summe	180 h
Berechnung des Workloads									
Präsenz	56 h								
Übungsbetrieb/Prüfungsvorbereitung	124 h								
Summe	180 h								
Turnus des Moduls	angeboten in jedem WiSe								
Voraussetzung für die Teilnahme	Keine <input type="checkbox"/> Folgende Inhaltliche Voraussetzungen: Praktische Informatik 2								
Lehr- und Lernformen	Seminar <input type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Tutorium <input checked="" type="checkbox"/> Praktikum <input type="checkbox"/> Projekt <input type="checkbox"/>								
Lernziele	<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>								
Lerninhalte	<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> <p>Lehrveranstaltung(en):</p> <ul style="list-style-type: none"> • 03-IBGP-PI3 Praktische Informatik 3: Funktionale Programmierung 								
Prüfungsformen	KP, PL1: xx%, PL2: xx%, Portfolio, Klausur								

Literatur

- Simon Thompson: Haskell - The Craft of Functional Programming, Addison-Wesley, 3. Auflage 2011.

Weiteres Lehrmaterial ist auf der Webseite der Veranstaltung zu finden:

- Folienkopien
- Übungsaufgaben
- Hinweise auf Quellen im WWW

Das Haskell-System ghci ist frei verfügbare Software (für Linux, Windows und MacOS).