| Praktische Informatik 3 | | Modulnummer: | | | |
|---|--|-----------------------------------|--|--|--|
| Practical Computer Science 3 | | IBGP-PI3 | | | |
| Bachelor Pflicht/Wahl ⊠ Wahl □ Basis □ Ergänzung □ Sonderfall □ | Zugeordnet zu Masterprofil Sicherheit und Qualität (SQ) KI, Kognition, Robotik (KIKR) Digitale Medien und Interaktion (DMI) | | | | |
| Modulbereich: Praktische und Technische Informatik | | | | | |
| Modulteilbereich: (keine Angabe) | | | | | |
| Anzahl der SWS $\begin{array}{ c c c c c c c c c c c c c c c c c c c$ | Kreditpunkte: 6 | Turnus angeboten in jedem WiSe | | | |
| Formale Voraussetzungen: - | | | | | |
| Inhaltliche Voraussetzungen: Praktische Informatik 2 | | | | | |
| Vorgesehenes Semester: 3. Semester | | | | | |
| Sprache: Deutsch | | | | | |
| 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. Die Vorlesung Praktische Informatik 3 vermittelt essenzielles Grundwissen und Basisfähigkeiten, deren Beherrschung für nahezu jede vertiefte Beschäftigung mit Informatik Voraussetzung ist. Inhalte: 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) – Listenkomprehen – zyklische Datenstrukturen 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 Im Übungsbetrieb; Programmentwicklung in Haskell — Realisierung einzelner, überschaubarer Programmieraufgaben in kleinen Gruppen Lehrveranstaltung(en): • 03-IBGP-PI3 Praktische Informatik 3: Funktionale Programmierung | | | | | |
| Unterlagen (Skripte, Literatur, Programme usw.): • Simon Thompson: Haskell - The Craft of Functional Programming, Addison-Wesley, 3. Auflage 2011. Weiteres Lehrmaterial ist auf der Webseite des Veranstaltung zu finden: • Folienkopien • Übungsaufgaben • Hinweise auf Quellen im WWW Das Haskell-System ghci ist frei verfügbare Software (für Linux, Windows und MacOS). | | | | | |
| Form der Prüfung: | | | | | |
| KP, PL1: xx%, PL2: xx%, Portfolio, Klausur | | | | | |

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