

Modulbezeichnung	<b>Praktische Informatik 1 (Kopie vom Tue May 26 14:16:10 +0200 2020) (Kopie vom Sun Jun 21 19:49:05 +0200 2020) (deleted:Mon Jun 22 11:07:15 +0200 2020)</b>								
Modulverantwortliche(r)	Dr. Thomas Röfer								
Modulart	Pflicht/Wahl <input checked="" type="checkbox"/> Wahlpflicht <input type="checkbox"/>								
Spezialisierungsbereich									
Dauer des Moduls	1 Semester								
Kreditpunkte	9 CP								
Arbeitsaufwand	<table> <tr> <td colspan="2">Berechnung des Workloads</td> </tr> <tr> <td>Präsenz</td> <td>112 h</td> </tr> <tr> <td>Übungsbetrieb/Prüfungsvorbereitung</td> <td>158 h</td> </tr> <tr> <td>Summe</td> <td>270 h</td> </tr> </table>	Berechnung des Workloads		Präsenz	112 h	Übungsbetrieb/Prüfungsvorbereitung	158 h	Summe	270 h
Berechnung des Workloads									
Präsenz	112 h								
Übungsbetrieb/Prüfungsvorbereitung	158 h								
Summe	270 h								
Turnus des Moduls	angeboten in jedem WiSe								
Voraussetzung für die Teilnahme	Keine <input checked="" type="checkbox"/> Folgende								
Lehr- und Lernformen	Seminar <input type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Tutorium <input checked="" type="checkbox"/> Praktikum <input checked="" type="checkbox"/> Projekt <input type="checkbox"/>								
Lernziele	<ul style="list-style-type: none"> <li>• Grundlegende Informatikkonzepte wiedergeben und erklären können.</li> <li>• Konzepte einer imperativen Programmiersprache kennen, verstehen und anwenden können.</li> <li>• Anschauliche Sachverhalte im Modell der Objektorientierung ausdrücken können.</li> <li>• Einfache Algorithmen entwickeln und in Java umsetzen können.</li> <li>• Einfache in Java realisierte Algorithmen systematisch testen können.</li> <li>• Probleme in Teilprobleme zerlegen und diese Strukturierung mit Mitteln von Java umsetzen und aussagekräftig dokumentieren können.</li> <li>• Formale Syntaxbeschreibungen verstehen und für einfache Sprachen entwickeln können.</li> <li>• Operationelle Semantik einfacher While-Sprachen verstehen und zum Nachweis einfacher Programmeigenschaften anwenden können</li> <li>• Eine Entwicklungsumgebung nutzen können.</li> <li>• LaTeX zur Erstellung einfacher Dokumente nutzen können.</li> <li>• Versionsverwaltungssysteme einsetzen können.</li> <li>• In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können.</li> </ul> <p>Die Vorlesungen Praktische Informatik 1 und 2 vermitteln essenzielles Grundwissen und Basisfähigkeiten, deren Beherrschung für nahezu jede vertiefte Beschäftigung mit Informatik – sowohl in der industriellen Anwendung, als auch in der Forschung – Voraussetzung ist.</p>								

Lerninhalte	<ol style="list-style-type: none"> <li>1. Basiswissen: von Neumannsche Rechnerorganisation – Grundlagen der Rechnerarchitektur – Programm und Prozess – Programmiersprachen – Compiler, Assembler, Loader, Linker, Interpreter, Laufzeitumgebungen, Betriebssysteme – Browser – Grafische Benutzungsschnittstellen – Shells</li> <li>2. Datenstrukturen: Information und ihre Repräsentation – Datentypen und Typanalyse – Elementare und zusammengesetzte Datentypen – rekursive Datentypen – Kanonische Operationen auf den eingeführten Datenstrukturen</li> <li>3. Algorithmen: Begriff des Algorithmus – Beschreibung von Algorithmen – Algorithmische Umsetzung kanonischer Operationen auf Datenstrukturen – Kontrollstrukturen – Rekursion – Grundlegende Strategien: Greedy-Strategie versus Divide-and-Conquer-Strategie</li> <li>4. Programmierparadigmen: (1) Imperative, funktionale und logische Programmierung, (2) Objektorientierte (imperative) Programmierung, (3) Sequenzielle Programme versus nebenläufige Programme</li> <li>5. Grundkomponenten imperativer Programmiersprachen: Schnittstellen und Ein-/Ausgabe, Variablen und Zuweisungen, Kontrollstrukturen, Blöcke, Funktionen, Rekursion</li> <li>6. Syntax und Semantik imperativer Programmiersprachen: Syntax und Methoden der Syntax-Spezifikation, reguläre Ausdrücke, (erweiterte) Backus-Naur-Form (E)BNF, Syntaxgraphen – operationelle Semantik für Zuweisungen und Kontrollstrukturen</li> <li>7. Prinzipien der objektorientierten Programmierung: Geheimnisprinzip – Methoden – Operationen – Objekte – Klassen – Botschaften – Ereignisverarbeitung – Attribute – Vererbung – Polymorphismus – Overloading</li> <li>8. Umsetzung der Punkte 2.-7. mit Java – Illustration anhand einfacher Algorithmen</li> <li>9. Programmdokumentation und zugehörige Hilfswerkzeuge, z.B. JavaDoc – Doxygen</li> <li>10. Testen von Programmen und zugehörige Hilfswerkzeuge, z.B. JUnit</li> <li>11. Basisdienste im Internet: telnet, ftp und ihre sicheren Varianten ssh, scp, sftp</li> <li>12. World-Wide-Web – Grundbegriffe von HTML</li> </ol> <p>Programmier-Praktikum: Programmentwicklung in Java – Realisierung einzelner, überschaubarer Programmieraufgaben</p> <p>Lehrveranstaltung(en):</p> <p>03-IBGP-PI1 Praktische Informatik 1: Imperative Programmierung und Objektorientierung</p>
Prüfungsformen	KP; PL1: 70
Literatur	<ul style="list-style-type: none"> <li>• G. Saake und K.-U. Sattler: Algorithmen und Datenstrukturen. dpunkt.verlag, Heidelberg (2004)</li> <li>• R. Schiedermeier: Programmieren mit Java. Pearson, München (2005)</li> </ul> <p>Weitere Informationen (Beispielprogramme, Musterlösungen, im WWW verfügbare Literatur) sind auf der Web-Seite der Veranstaltung zu finden.</p>