Modulbezeichnung	Praktische Informatik 2
Modulverantwortliche(r)	Prof. Dr. U. Bormann
Modulart	Pflicht/Wahl ⊠ Wahlpflicht □
Spezialisierungsbereich	
Dauer des Moduls	1 Semester
Kreditpunkte	6 CP
Arbeitsaufwand	Berechnung des Workloads Präsenz 56 h Übungsbetrieb/Prüfungsvorbereitung 124 h Summe 180 h
Turnus des Moduls	angeboten in jedem SoSe
Voraussetzung für die Teilnahme	Keine ⊠  Folgende Inhaltliche Voraussetzungen: Praktische Informatik 1
Lehr- und Lernformen	Seminar □ Vorlesung ⊠ Tutorium ⊠ Praktikum □ Projekt □
Lernziele	<ul> <li>Typische Datenstrukturen identifizieren und problemadäquat einsetzen können.</li> <li>Datenstrukturen und Algorithmen in Java umsetzen können.</li> <li>Wesentliche Algorithmen der Informatik erklären, anwenden und modifizieren können.</li> <li>Algorithmische Alternativen bezüglich der Eignung für ein Problem beurteilen können.</li> <li>Grundbegriffe der formalen Verifikation erläutern können.</li> <li>Die Komplexität von einfachen Algorithmen analysieren können.</li> <li>Eine komplexe Entwicklungsumgebung nutzen können.</li> <li>Generische und funktionale Konzepte in eigenen Programmen einsetzen können.</li> <li>In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können.</li> <li>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.</li> </ul>

1 Algorithman Boariff des Algorithmus Boarbailtung von Algorithman Algorithmicale Ilmostrung
<ol> <li>Algorithmen: Begriff des Algorithmus – Beschreibung von Algorithmen – Algorithmische Umsetzung kanonischer Operationen auf Datenstrukturen – Grundlegende Strategien: Greedy, Divide-and-Conquer, Backtracking, dynamische Programmierung</li> </ol>
2. Komplexität von Algorithmen – O(n)-Notation und asymptotische Analyse
<ol> <li>Suchen und Sortieren auf Arrays: Binäre Suche – Quicksort und weitere Sortieralgorithmen – Komplexitätsvergleiche</li> </ol>
<ol> <li>Mengen – Multimengen – Relationen – Funktionen: Datenstrukturen und Algorithmen zur Realisierung kanonischer Operationen (z.B. Mengenalgebra)</li> </ol>
<ol> <li>Listen – Stapel – Warteschlangen: Datenstrukturen zur Realisierung (Arrays versus Verkettung und dynamische Speicherallokation für Elemente), Algorithmen zur Realisierung kanonischer Operationen (Listentraversion, Anfügen, Einfügen, Löschen, Suchen, Stack-Operationen, FIFO-Warteschlangenoperationen)</li> </ol>
6. Bäume: Binäre Bäume, AVL-Bäume, Rot-Schwarz-Bäume, B-Bäume – Suchen, Einfügen, Löschen, Traversion
7. Hashing: Hash-Array, Hashfunktion, Hash Buckets, offenes Hashing
8. Graphen: ungerichtete, gerichtete, gewichtete Graphen – Repräsentation durch Knoten- und Kantenlisten, durch Adjazenzmatrizen, Adjazenzlisten – Algorithmen auf Graphen: Breitensuche, Tiefensuche, kürzeste Wege auf gewichteten Graphen: Dijkstras Algorithmus, minimal aufspannende Bäume: Algorithmen von Prim et al. und Kruskal
9. Spezifikation von Programmen: Vor- und Nachbedingungen – Invarianten
<ol> <li>Verifikation: Partielle und totale Korrektheit sequenzieller Programme – Formale Verifikation, z.B.</li> <li>Hoare Logik (Pre-/Postconditions) – Eigenschaftsbeweis durch Strukturelle Induktion</li> </ol>
nrveranstaltung(en):
03-IBGP-PI2 Praktische Informatik 2: Algorithmen und Datenstrukturen
PL1: 70%, PL2: 30%, Portfolio, Klausur
G. Saake und KU. Sattler: Algorithmen und Datenstrukturen. dpunkt.verlag, Heidelberg (2004)
• R. Schiedermeier: Programmieren mit Java. Pearson, München (2005)
itere Informationen (Beispielprogramme, Musterlösungen, im WWW verfügbare Literatur) sind auf der b-Seite der Veranstaltung zu finden.