Modulbezeichnung	Praktische Informatik 2
Modulverantwortliche(r)	Dr. K. Hölscher
Modulart	Pflicht/Wahl ⊠ Wahlpflicht □
Spezialisierungsbereich	
Dauer des Moduls	1 Semester
Kreditpunkte	9 CP
Arbeitsaufwand	Berechnung des Workloads Präsenz 84 h Übungsbetrieb/Prüfungsvorbereitung 186 h Summe 270 h
Turnus des Moduls	angeboten in jedem SoSe
Voraussetzung für die Teilnahme	Keine ⊠ Folgende Inhaltliche Voraussetzungen: Grundlagen der Programmierung
Lehr- und Lernformen	Seminar □ Vorlesung ⊠ Tutorium ⊠ Praktikum □ Projekt □
Lernziele	 Grundlegende Konzepte der objektorientierten Programmierung kennen, verstehen und anwenden können. Anschauliche Sachverhalte im Modell der Objektorientierung ausdrücken können. Eine einfache Entwicklungsumgebung nutzen können. LaTeX zur Erstellung einfacher Dokumente nutzen können. Versionsverwaltungssysteme verstehen und einsetzen können. Datenstrukturen und Algorithmen in Java umsetzen können Fehler unter Einsatz eines einfachen Debuggers finden können. Einfache Komponententests zur Qualitätssicherung erstellen und durchführen können. Ein Softwaredokumentationswerkzeug verwenden können. Typische Datenstrukturen identifizieren und problemadäquat einsetzen können. Wesentliche Algorithmen der Informatik erklären, anwenden und modifizieren können. Algorithmische Alternativen bezüglich der Eignung für ein Problem beurteilen können. Die Komplexität von einfachen Algorithmen analysieren können. In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können.

	Prinzipien der objektorientierten Programmierung: Geheimnisprinzip – Methoden – Operationen – Objekte – Klassen – Botschaften – Ereignisverarbeitung – Attribute – Vererbung – Polymorphismus – Überladung – Generische Datentypen – Interfaces
	Datenstrukturen: Information und ihre Repräsentation – Datentypen und Typanalyse – Elementare und zusammengesetzte Datentypen – rekursive Datentypen
	3. Fehlervermeidung: Exceptions
	4. Dokumentation von Klassen, Methoden und Attributen
	5. Automatisierte Komponententests
	6. Fehlersuche (Debugging): Breakpoint – schrittweise Ausführung – Stacktrace
	7. Umsetzung der Punkte 16. mit Java, Javadoc und JUnit
	8. Algorithmen: Begriff des Algorithmus – Beschreibung von Algorithmen – Algorithmische Umsetzung kanonischer Operationen auf Datenstrukturen – Grundlegende Strategien: Greedy, Divide-and-Conquer, Backtracking, dynamische Programmierung, zufallsgesteuerte Algorithmen, genetische Algorithmen, heuristische Algorithmen, probabilistische Algorithmen
	9. Komplexität von Algorithmen – O(n)-Notation und asymptotische Analyse
Lerninhalte	 Suchen und Sortieren auf Arrays: Binäre Suche – Quicksort und weitere Sortieralgorithmen – Komplexitätsvergleiche
	11. 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)
	12. Bäume: Binäre Bäume, AVL-Bäume, Rot-Schwarz-Bäume, B-Bäume – Suchen, Einfügen, Löschen, Traversion
	13. Hashing: Hash-Array, Hashfunktion, Hash Buckets, offenes Hashing
	14. 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
	Im Rahmen des Übungsbetriebes werden LATEX und Versionskontrolle mittels Git eingeführt und verwendet.
	Lehrveranstaltung(en):
	03-B-MI-22.1 Objektorientierte Programmierung [OOP] (3 CP)
	03-B-MI-22.2 Algorithmen und Datenstrukturen [AuD] (6 CP)
Prüfungsformen	KP; PL1: 30%, PL2: 55%, PL3: 15% ; Klausur, Portfolio, Fachgespräch
Literatur	 David J. Barnes, Michael Kölling: Java lernen mit BlueJ - Objects first - Eine Einführung in Java. Aktuelle Auflage. Pearson Studium.
	Christian Ullenboom: Java ist auch eine Insel. Aktuelle Auflage. Rheinwerk Computing.
	 Thomas Ottmann, Peter Widmayer: Algorithmen und Datenstrukturen. Aktuelle Auflage, Spektrum Akademischer Verlag.
	Robert Sedgewick, Robert Wayne: Algorithmen. Aktuelle Auflage. Pearson Studium.
	Markus von Rimscha: Algorithmen kompakt und verständlich. Aktuelle Auflabe. Springer Vieweg.