

Praktische Informatik 2 <i>Practical Computer Science 2</i>								Modulnummer: INF-2	
Bachelor					Schwerpunkt				
Pflicht <input checked="" type="checkbox"/>					Computational Finance <input type="checkbox"/>				
Winf-Schwerpunkt-Pflicht <input type="checkbox"/>					E-Business <input type="checkbox"/>				
Winf-Schwerpunkt-Wahlpflicht <input type="checkbox"/>					IT-Management <input type="checkbox"/>				
Winf-Wahl <input type="checkbox"/>					Logistik <input type="checkbox"/>				
Anzahl der SWS	V	UE	K	S	Prak.	Proj.	Σ	Kreditpunkte: 9	
	0	0	6	0	0	0	6	Turnus angeboten in jedem SoSe	
Formale Voraussetzungen: -									
Inhaltliche Voraussetzungen: Grundlagen der Programmierung									
Vorgesehenes Semester: 2. Semester									
Sprache: Deutsch									
Ziele: <ul style="list-style-type: none"> • 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. 									

Inhalte:

1. Prinzipien der objektorientierten Programmierung: Geheimnisprinzip – Methoden – Operationen – Objekte – Klassen – Botschaften – Ereignisverarbeitung – Attribute – Vererbung – Polymorphismus – Überladung – Generische Datentypen – Interfaces
2. 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 1.-6. 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
10. 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 (Listentraversal, Anfügen, Einfügen, Löschen, Suchen, Stack-Operationen, FIFO-Warteschlangenoperationen)
12. Bäume: Binäre Bäume, AVL-Bäume, Rot-Schwarz-Bäume, B-Bäume – Suchen, Einfügen, Löschen, Traversal
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)

Unterlagen (Skripte, Literatur, Programme usw.):

- 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 Auflage. Springer Vieweg.

Form der Prüfung:

KP; PL1: 30%, PL2: 55%, PL3: 15% ; Klausur, Portfolio, Fachgespräch

Arbeitsaufwand	Präsenz	84 h
	Übungsbetrieb/Prüfungsvorbereitung	186 h
	Summe	270 h

Lehrende:
Dr. K. Hölscher

Verantwortlich:
Dr. K. Hölscher