

Praktische Informatik 1 <i>Practical Computer Science 1</i>								Modulnummer: INF-1	
Bachelor Pflicht <input checked="" type="checkbox"/> Winf-Schwerpunkt-Pflicht <input type="checkbox"/> Winf-Schwerpunkt-Wahlpflicht <input type="checkbox"/> Winf-Wahl <input type="checkbox"/>					Schwerpunkt Computational Finance <input type="checkbox"/> E-Business <input type="checkbox"/> IT-Management <input type="checkbox"/> Logistik <input type="checkbox"/>				
Anzahl der SWS	V	UE	K	S	Prak.	Proj.	Σ	Kreditpunkte: 8	Turnus angeboten in jedem WiSe
	4	0	0	0	4	0	8		
Formale Voraussetzungen: -									
Inhaltliche Voraussetzungen: -									
Vorgesehenes Semester: 1. Semester									
Sprache: Deutsch									
Ziele: <ul style="list-style-type: none"> • Grundlegende Informatikkonzepte wiedergeben und erklären können. • Konzepte einer imperativen Programmiersprache kennen, verstehen und anwenden können. • Anschauliche Sachverhalte im Modell der Objektorientierung ausdrücken können. • Einfache Algorithmen entwickeln und in Java umsetzen können. • Einfache in Java realisierte Algorithmen systematisch testen können. • Probleme in Teilprobleme zerlegen und diese Strukturierung mit Mitteln von Java umsetzen und aussagekräftig dokumentieren können. • Formale Syntaxbeschreibungen verstehen und für einfache Sprachen entwickeln können. • Operationelle Semantik einfacher While-Sprachen verstehen und zum Nachweis einfacher Programmeigenschaften anwenden können • Eine Entwicklungsumgebung nutzen können. • LaTeX zur Erstellung einfacher Dokumente nutzen können. • Versionsverwaltungssysteme einsetzen können. • In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können. 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.									

Inhalte:

1. Basiswissen: von Neumannsche Rechnerorganisation – Grundlagen der Rechnerarchitektur – Programm und Prozess – Programmiersprachen – Compiler, Assembler, Loader, Linker, Interpreter, Laufzeitumgebungen, Betriebssysteme – Browser – Grafische Benutzungsschnittstellen – Shells
2. Datenstrukturen: Information und ihre Repräsentation – Datentypen und Typanalyse – Elementare und zusammengesetzte Datentypen – rekursive Datentypen – Kanonische Operationen auf den eingeführten Datenstrukturen
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
4. Programmierparadigmen: (1) Imperative, funktionale und logische Programmierung, (2) Objektorientierte (imperative) Programmierung, (3) Sequenzielle Programme versus nebenläufige Programme
5. Grundkomponenten imperativer Programmiersprachen: Schnittstellen und Ein-/Ausgabe, Variablen und Zuweisungen, Kontrollstrukturen, Blöcke, Funktionen, Rekursion
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
7. Prinzipien der objektorientierten Programmierung: Geheimnisprinzip – Methoden – Operationen – Objekte – Klassen – Botschaften – Ereignisverarbeitung – Attribute – Vererbung – Polymorphismus – Overloading
8. Umsetzung der Punkte 2.-7. mit Java – Illustration anhand einfacher Algorithmen
9. Programmdokumentation und zugehörige Hilfswerkzeuge, z.B. JavaDoc – Doxygen
10. Testen von Programmen und zugehörige Hilfswerkzeuge, z.B. JUnit
11. Basisdienste im Internet: telnet, ftp und ihre sicheren Varianten ssh, scp, sftp
12. World-Wide-Web – Grundbegriffe von HTML

Programmier-Praktikum: Programmentwicklung in Java – Realisierung einzelner, überschaubarer Programmieraufgaben

Unterlagen (Skripte, Literatur, Programme usw.):

- G. Saake und K.-U. Sattler: Algorithmen und Datenstrukturen. dpunkt.verlag, Heidelberg (2004)
- R. Schiedermeier: Programmieren mit Java. Pearson, München (2005)

Weitere Informationen (Beispielprogramme, Musterlösungen, im WWW verfügbare Literatur) sind auf der Web-Seite der Veranstaltung zu finden.

Form der Prüfung:

i.d.R. Bearbeitung von Übungsaufgaben und Klausur

Arbeitsaufwand	Präsenz	112 h
	Übungsbetrieb/Prüfungsvorbereitung	128 h
	Summe	240 h

Lehrende:
Dr. T. Röfer, Dr. K. Hölscher

Verantwortlich:
Dr. Thomas Röfer