Algorithmen-Design

Stufenweises Vorgehen vom Problem zur Lösung (mit Beispiel): Problembeschreibung, Modellierung, Algorithmen-Design, Algorithmen-Analyse, Implementierung und Test Tiefensuche in gerichteten und ungerichteten Graphen. Zusammenhangskomponenten, Breitensuche, kürzeste Wege Inhalte Allgemeine Muster für den Entwurf von Algorithmen, jeweils mit Anwendungsbeispielen: Exhaustive Search, Backtracking, Branch-and-Bound, Dynamische Programmierung, Divide-and-Conquer, lokale Suche Laufzeitanalysen, Verfahren zur Lösung von Rekursionsgleichungen Die Studierenden sollen algorithmische Aufgabenstellungen verstehen und formulieren können. Algorithmen für vorgegebene Aufgabestellungen durch Einsatz der Algorithmen-Entwurfsmuster entwickeln Lernziele können, Algorithmen hinsichtlich Korrektheit und Laufzeit analysieren können, Einsatz und Bedeutung effizienter Algorithmen in verschiedenen Anwendungsbereichen kennen lernen. Vorlesung (2 SWS) Übung (1 SWS) Lehrform Seminar/Seminaristischer Unterricht **Art und Umfang** Labor (1 SWS) **Projekt** Beherrschung des Lehrstoffes der Module "Theoretische Voraussetzungen für die Informatik", "Objektorientierte Programmierung" und **Teilnahme** "Datenstrukturen und Algorithmen" Regelmäßige Teilnahme an der Vorlesung Regelmäßige Teilnahme an den Übungen Prüfungsvorleistung Regelmäßige Bearbeitung von Haus-/Laborarbeiten Bestehen von Leistungsstandkontrollen Schriftliche Prüfung (90 Minuten Bearbeitungszeit) Prüfungsform

Mündliche Prüfung

Prüfung am PC

Art und Umfang

	☐ Hausarbeit/Projekt mit Kolloquium	
	Informatik	▼ _{PF} □ WPF
	Informatik - Digitale Medien und Spiele (Schwerpunkt Medien)	□ _{PF} ▼ WPF
Verwendbarkeit	Informatik - Digitale Medien und Spiele (Schwerpunkt Spiele)	□ _{PF} ▼ WPF
	Informatik - Sichere und mobile Systeme	□ _{PF} ▼ WPF
	Medizininformatik	□ _{PF} ▼ WPF
Angebot	Sommersemester Wintersemester U	nregelmäßig
Arbeitsaufwand	ECTS-Punkte Kontaktzeit Selbststudium 5 60 Stunden 90 Stunden	
Lehrende(r)	Prof. Dr. H. Schmitz	
Modulver antwortliche(r)	Prof. Dr. H. Schmitz	
Änderungsdatum	04.07.2013	

Datenstrukturen und Algorithmen

Inhalte	 Einführung in die wichtigsten Datenstrukturen von Programmiersprachen Sequenzen, Listen Stacks, Queues Hashing Binäre Bäume, Heaps Einführung in grundlegende Such- und Sortierverfahren Laufzeit- und Speicherplatzbetrachtungen
Lernziele	 die grundlegenden Datenstrukturen und Algorithmen kennen lernen, deren Wechselwirkungen, insbesondere unter Laufzeit- und Speicherplatzbetrachtungen, verstehen und auf praktische Beispiele anwenden können, die wesentlichen Such- und Sortieralgorithmen verstehen und nach Anwendungsszenarien beurteilen und auswählen können, die grundlegenden Datenstrukturen und Algorithmen in Java umsetzen können.
Lehrform Art und Umfang	 ✓ Vorlesung (2 SWS) ✓ Übung (2 SWS) ✓ Seminar/Seminaristischer Unterricht ✓ Labor ✓ Projekt
Voraussetzungen für die Teilnahme	Beherrschung des Lehrstoffes des Moduls "Objektorientierte Programmierung"
Prüfungsvorleistung	 □ Regelmäßige Teilnahme an der Vorlesung ☑ Regelmäßige Teilnahme an den Übungen □ Regelmäßige Bearbeitung von Haus-/Laborarbeiten ☑ Bestehen von Leistungsstandkontrollen
Prüfungsform Art und Umfang	 ✓ Schriftliche Prüfung (60 Minuten Bearbeitungszeit) ✓ Mündliche Prüfung ✓ Prüfung am PC ✓ Hausarbeit/Projekt mit Kolloquium (Bearbeitung

	vorlesungsbegleitend)	
	Informatik	▼ _{PF} □
Verwendbarkeit	Informatik - Digitale Medien und Spiele (Schwerpunkt Medien)	PF WPF
	Informatik - Digitale Medien und Spiele (Schwerpunkt Spiele)	PF WPF
	Informatik - Sichere und mobile Systeme	PF WPF
	Medizininformatik	PF WPF
Angebot	Sommersemester Wintersemester Unregelm	ıäßig
Arbeitsaufwand	ECTS-Punkte Kontaktzeit Selbststudium	
	5 60 Stunden 90 Stunden	
Lehrende(r)	Prof. Dr. A. Lux	
Modulverantwortliche(r)	Prof. Dr. A. Lux	
Änderungsdatum	19 01 2017	

Objektorientierte Programmierung

- Einführung in die Programmierung
 - Einführung in die Informatik: Geschichte, Begriffsklärung, Teilgebiete
 - Formale Sprachbeschreibung: Grammatiken, Syntaxdiagramme
 - o Algorithmen: Vom Problem zum Programm
 - Sequenzen als Beispiel einer Datenstruktur
- Programmieren in Java
 - Das Java-System
 - o Grundelemente von Java
 - Operatoren und Ausdrücke
 - o Anweisungen
 - Einführung in die Objektorientierte Programmierung
 - Grundkonzepte der Objektorientierten Programmierung
 - Klassen und Objekte
 - Module
 - Vererbung
 - o Zeichenketten und Felder
 - o Ausnahmebehandlung

Die Studierenden sollen

- einen Überblick über Geschichte und Entwicklung des Fachgebietes Informatik erhalten
- den Aufbau und die Beschreibung von Programmiersprachen als formale Sprachen lernen
- einfache Algorithmen und Datenstrukturen kennen und formal beschreiben können
- unterschiedliche Paradigmen von Programmiersprachen kennen lernen
- alle wichtigen Elemente der Programmiersprache Java anwenden können
- einfache Java-Programme analysieren und erstellen können
- die Grundkonzepte der objektorientierten Programmierung verstehen und mit der Standardnotation UML beschreiben können
- die Prinzipien der objektorientierten Programmierung in Java umsetzen können

	Vorlesung (4 SWS)
Lehrform Art und Umfang	Übung (4 SWS)
	\square Seminar/Seminaristischer Unterricht
	Labor

Inhalte

Lernziele

	Projekt	
Voraussetzungen für die Teilnahme	Keine	
Prüfungsvorleistung	 □ Regelmäßige Teilnahme an der Vorlesung □ Regelmäßige Teilnahme an den Übungen □ Regelmäßige Bearbeitung von Haus-/Laborarbeiten □ Bestehen von Leistungsstandkontrollen 	
Prüfungsform Art und Umfang	 ✓ Schriftliche Prüfung (90 Minuten Bearbeitungszeit) ✓ Mündliche Prüfung ✓ Prüfung am PC ✓ Hausarbeit/Projekt mit Kolloquium 	
	Informatik	▼ _{PF} □ WPF
	Informatik - Digitale Medien und Spiele (Schwerpunkt Medien)	✓ PF WPF
Verwendbarkeit	Informatik - Digitale Medien und Spiele (Schwerpunkt Spiele)	PF WPF
	Informatik - Sichere und mobile Systeme	✓ PF WPF
	Medizininformatik	✓ PF WPF
Angebot	Sommersemester Wintersemester Unregelm	näßig
Arbeitsaufwand	ECTS-Punkte Kontaktzeit Selbststudium	
	10 120 Stunden 180 Stunden	
Lehrende(r)	Prof. Dr. A. Künkler, Prof. Dr. A. Lux, Prof. Dr. G. Schneide Rock	er, Prof. Dr. G.

Modulverantwortliche(r) Prof. Dr. G. Schneider

Änderungsdatum 11.07.2013

Programmierparadigmen

- Überblick über Programmierparadigmen
- Grundkonzepte der funktionalen Programmierung in LISP und/oder Haskell
- Funktionale Abstraktion

Inhalte

Lernziele

- Funktionen h\u00f6herer Ordnung
- Rekursion und algebraische Datenstrukturen
- Symbolische Repräsentation und Verarbeitung
- Funktionale Konzepte in modernen Sprachen und Frameworks (Java 8 Lambdas und Streams, Big Data, Reactive Programming)

Studierende sollten nach der Veranstaltung in der Lage sein,

- Probleme funktional zu abstrahieren, zu beschreiben und zu lösen
- rekursive Algorithmen und Datenstrukturen zu entwerfen
- Funktionen höherer Ordnung zu erkennen und anzuwenden
- funktionale Konzepte in anderen Programmiersprachen und Frameworks, z. B. Java Streams, zu verstehen und effektiv zu nutzen

Lehrform Art und Umfang	✓ Vorlesung (2 SWS)
	☑ Übung (2 SWS)
	Seminar/Seminaristischer Unterricht
	Labor
	Projekt
oraussetzungen für die Teilnahme	Keine
Prüfungsvorleistung	 Regelmäßige Teilnahme an der Vorlesung ✓ Regelmäßige Teilnahme an den Übungen ✓ Regelmäßige Bearbeitung von Haus-/Laborarbeiten Destehen von Leistungsstandkontrollen
Prüfungsform Art und Umfang	 ✓ Schriftliche Prüfung (90 Minuten Bearbeitungszeit) ✓ Mündliche Prüfung ✓ Prüfung am PC ✓ Hausarbeit/Projekt mit Kolloquium

Verwendbarkeit	Informatik	✓ PF WPF	
	Informatik - Digitale Medien und Spiele (Schwerpunkt Medien)	□ _{PF} ✓	
	Informatik - Digitale Medien und Spiele (Schwerpunkt Spiele)	□ _{PF} ▼	
	Informatik - Sichere und mobile Systeme	□ _{PF} ✓	
	Medizininformatik	□ _{PF} ▼	
Angebot	Sommersemester Wintersemester Unregelmäßig		
Arbeitsaufwand	ECTS-Punkte Kontaktzeit Selbststudium		
	5 60 Stunden 90 Stunden		
Lehrende(r)	Prof. Dr. Christoph Schmitz		
Modulverantwortliche(r)	Prof. Dr. Christoph Schmitz		
Änderungsdatum	18 10 2017		

Softwareentwurf und -test

- Software-Entwurf
 - Einführung in die Objektorientierte Softwareentwicklung
 - Konzepte und Notation (UML) für die objektorientierte Analyse
 - Grundkonzepte
 - Statische Konzepte
 - o Dynamische Konzepte
 - o Analysemuster
 - o Checklisten zur Erstellung eines OOA-Modells
 - Konzepte und Notation (UML) für den objektorientierten Entwurf
 - Entwurfsmuster
- Software-Test
 - Bedeutung der Software-Prüfung, Sicherheit, Kosten, psychologische Aspekte
 - o Review-Verfahren
 - o Grundbegriffe und Modellbildung beim Testen
 - o Verfahren zur Aufstellung von Testfällen
 - o Testen im Software-Entwicklungsprozess
 - Besonderheiten beim Test objektorientierter Programme

Die Studierenden sollen

- die objektorientierten Konzepte in den Phasen Analyse und Entwurf anwenden können,
- die objektorientierten Konzepte mit der Standardnotation UML beschreiben können,
- wissen, wie Sie am besten beim Erstellen objektorientierter Modelle vorgehen und wie sie gute von schlechten Modellen unterscheiden können,
- die objektorientierten Konzepte in Java umsetzen können,
- Analyse- und Entwurfsmuster anwenden können.
- die Bedeutung der Software-Prüfung kennen lernen,
- die wesentlichen Testverfahren verstehen und anwenden können.

Lehrform Art und Umfang	✓ Vorlesung (2 SWS)
	☑ Übung (2 SWS)
	\square Seminar/Seminaristischer Unterricht
	Labor
	Projekt

Inhalte

Lernziele

Voraussetzungen für die	Beherrschung des Lehrstoffes der Module "Objektorientierte	
Teilnahme	Programmierung" und "Grundlagen der Mathematik"	
	Regelmäßige Teilnahme an der Vorlesung	
Prüfungsvorleistung	Regelmäßige Teilnahme an den Übungen	
0 0	Regelmäßige Bearbeitung von Haus-/Laborarbeiten	
	Bestehen von Leistungsstandkontrollen	
	Schriftliche Prüfung (90 Minuten Bearbeitungszeit)	
Prüfungsform	Mündliche Prüfung	
Art und Umfang	Prüfung am PC	
	Hausarbeit/Projekt mit Kolloquium	
	Informatik	▼ _{PF} □
	Informatik - Digitale Medien und Spiele (Schwerpunkt Medien)	PF WPF
Verwendbarkeit	Informatik - Digitale Medien und Spiele (Schwerpunkt Spiele)	PF WPF
	Informatik - Sichere und mobile Systeme	PF WPF
	Medizininformatik	PF WPF
Angebot	Sommersemester Wintersemester Unregelm	äßig
	ECTS-Punkte Kontaktzeit Selbststudium	
Arbeitsaufwand	5 60 Stunden 90 Stunden	
Lehrende(r)	Prof. Dr. A. Künkler	
Modulverantwortliche(r)	Prof. Dr. A. Künkler	
Änderungsdatum	24.04.2015	