

# Modulhandbuch

## M.Sc. Informatik

FernUniversität in Hagen  
Fakultät für Mathematik und Informatik

Stand:  
30.11.2021

# Inhaltsverzeichnis

<b>Katalog B</b>	3
<b>Katalog M1: Formale Methoden und Algorithmik</b>	31
<b>Katalog M2: Computersysteme</b>	41
<b>Katalog M3: Informationssysteme</b>	56
<b>Katalog M4: Softwaretechnik und Programmierung</b>	76
<b>Fachpraktika</b>	81
<b>Masterseminar und Einführung in das wissenschaftliche Arbeiten</b>	110
<b>Abschlussmodul</b>	113
<i>Detailliertes Inhaltsverzeichnis</i>	116

# Katalog B

Lehrende/r	Winfried Hochstättler	Modulbeauftragte/r	Winfried Hochstättler
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01212 Lineare Optimierung		SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 18 Stunden): 126 Stunden Einüben des Stoffes (insbesondere durch Einsendeaufgaben (7 mal 15 Stunden): 105 Stunden Wiederholung und Prüfungsvorbereitung (Studientag und Selbststudium): 69 Stunden		
Qualifikationsziele	Die Studierenden können lineare Optimierungsaufgaben modellieren, in Normalformen bringen und dualisieren. Sie kennen Polyedertheorie als Geometrie der linearen Optimierung. Sie kennen die Algebra und die Geometrie des Simplexverfahrens und zugehörige komplexitätstheoretische Überlegungen zur Linearen Optimierung. Sie kennen Bedeutung und Vorgehensweise der Ellipsoidmethode und von Innere-Punkt-Verfahren.		
Inhalte	Zunächst stellen wir die Aufgabenstellung vor, modellieren verschiedene Probleme als Lineares Programm und lösen diese mit Standardsoftware. Dann stellen wir die Dualitätstheorie mitsamt der zugehörigen Linearen Algebra vor. Im Folgenden analysieren wir die Seitenflächenstruktur von Polyedern und diskutieren das Simplexverfahren, seine Varianten und zugehörige Komplexitätsuntersuchungen. Weiter diskutieren wir die Ellipsoidmethode und ihre Bedeutung für die kombinatorische Optimierung sowie das Karmarkar-Verfahren und Innere-Punkt-Methoden.		
	<b>Ergänzende Literatur:</b> B. Gärtner, J. Matousek: Understanding and Using Linear Programming, Springer-Verlag, 2006 G. M. Ziegler: Polyhedral Theory A. Schrijver: Theory of Linear and Integer Programming, WILEY, 1998 C. Roos, T. Terlaky, J.-P. Vial: Interior Point Methods for Linear Optimization, Springer-Verlag, 2005		
Inhaltliche Voraussetzung	Das Modul setzt die Module 61111 "Mathematische Grundlagen" (01141), 61211 "Analysis" (01144) und insbesondere sehr gute Kenntnisse des Moduls 61112 "Linearen Algebra" (01143) voraus.		
Lehr- und Betreuungsformen	Kursmaterial Studientag/e internetgestütztes Diskussionsforum Zusatzmaterial Betreuung und Beratung durch Lehrende Einsendeaufgaben mit Korrektur und/oder Musterlösung		
Anmerkung	-		
Formale Voraussetzung	keine		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematik B.Sc. Mathematisch-technische Softwareentwicklung		

M.Sc. Informatik  
M.Sc. Praktische Informatik

**Prüfungsformen**

Art der Prüfungsleistung

Voraussetzung

Prüfung  
Stellenwert 1/12  
der Note

bestandene benotete mündliche  
Modulprüfung

keine

## Lehrende/r

André Schulz

Modulbeauftragte/r

Friedrich Steimann

Friedrich Steimann

Dauer des Moduls  
ein SemesterECTS  
10Workload  
300 StundenHäufigkeit  
in jedem Wintersemester

## Lehrveranstaltung(en)

01810 Übersetzerbau

WS

SWS  
4+2

## Detaillierter Zeitaufwand

Bearbeiten der Kurseinheiten: 150 Stunden

Bearbeitung der Einsendeaufgaben: 75 Stunden

Wiederholung und Prüfungsvorbereitung, Prüfung: 75 Stunden

## Qualifikationsziele

Nach erfolgreicher Teilnahme besitzen Studierende einen guten Überblick über die Techniken des Übersetzerbaus. Sie können erklären, mit welchen Schritten höhere Programmiersprachen in Maschinensprache überführt werden. Sie sind in der Lage, eigene kleine Programmiersprachen für spezielle Anwendungen, komplexe Dateiformate oder Protokolle für die Client-Server-Kommunikation zu definieren und dafür mit Hilfe von Werkzeugen wie Lex/Yacc (Flex/Bison) entsprechende lexikalische Analysatoren und Parser zu konstruieren. Sie können Definitionen für wesentliche Begriffe der zugrundeliegenden Theorie angeben, etwa für kontextfreie Grammatiken, reguläre Sprachen/Ausdrücke, endliche Automaten, attributierte Grammatiken. Sie können die Konstruktion und Arbeitsweise von Top-Down- oder Bottom-Up-Parsern im Detail erklären.

## Inhalte

Der Kurs behandelt Techniken und Werkzeuge zur Übersetzung einer (formalen) Quellsprache in eine Zielsprache. Zumindest die Quellsprache ist durch eine geeignete Grammatik definiert. Der klassische Anwendungsfall ist die Übersetzung von einer höheren Programmiersprache in Maschinen- oder Assemblersprache. Übersetzungstechniken haben aber viel weitergehende Anwendbarkeit: Andere Quellsprachen können z.B. Dokumentbeschreibungssprachen sein (LaTeX, HTML), Anfragesprachen in Datenbanksystemen, VLSI-Entwurfssprachen usw. oder auch komplex strukturierte Eingabedateien für Anwendungsprogramme. Es gibt mächtige Werkzeuge (Scanner- und Parsergeneratoren auf der Basis attributierter Grammatiken), deren Kenntnis für jeden Informatiker nützlich ist, auch wenn nur wenige klassische Übersetzer (Compiler) implementieren.

Inhalte: Lexikalische Analyse (Scanner), Syntaxanalyse (Parser), Syntaxgesteuerte Übersetzung, Übersetzung einer Dokument-Beschreibungssprache, Übersetzung imperativer Programmiersprachen, Übersetzung funktionaler Programmiersprachen, Codeerzeugung und Optimierung.

In den Übungen wird u.a. ein größeres durchgängiges Projekt zum Compilerbau bearbeitet.

## Ergänzende Literatur:

A.V. Aho, M.S. Lam, R. Sethi, J.D. Ullman: Compilers: Principles, Techniques, and Tools. 2. Aufl. Addison-Wesley, 2006.

R. Wilhelm, D. Maurer, Übersetzerbau: Theorie, Konstruktion, Generierung. 2. Aufl., Springer-Verlag, 2007.

R.H. Güting, M. Erwig: Übersetzerbau: Techniken, Werkzeuge, Anwendungen. Springer-Verlag, 1999.

Inhaltliche  
Voraussetzung

Grundbegriffe der Theorie der formalen Sprachen sind nützlich, werden aber auch in diesem Kurs vermittelt.

Lehr- und  
Betreuungsformen

Kursmaterial  
Einsendeaufgaben mit Korrektur und/oder Musterlösung  
internetgestütztes Diskussionsforum  
Betreuung und Beratung durch Lehrende

Anmerkung

-

Formale Voraussetzung

keine

Verwendung des Moduls

B.Sc. Informatik  
B.Sc. Mathematisch-technische Softwareentwicklung  
B.Sc. Wirtschaftsinformatik  
M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/12	Modulprüfung	

63117

## Data Mining

Lehrende/r	Fabio Valdés	Modulbeauftragte/r	Fabio Valdés
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01660 Data Mining: Konzepte und Techniken		SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 160 Stunden Bearbeitung der Einsendeaufgaben inkl. Verarbeitung des Korrektur-Feedbacks: 80 Stunden Wiederholung und Prüfungsvorbereitung, Prüfung: 60 Stunden		
Qualifikationsziele	Nach erfolgreicher Teilnahme besitzen die Studierenden einen umfassenden Überblick zu Wissensentdeckungsprozessen in Datenmengen/-banken. Sie sind in der Lage, verschiedene Attributtypen zu beschreiben und zu visualisieren sowie entsprechende Abstandsmaße zu berechnen. Sie besitzen detaillierte Kenntnisse zur Datenvorverarbeitung. Sie sind mit den Zielen und Methoden der zentralen Data-Mining-Techniken Mustersuche, Klassifikation und Clusteranalyse vertraut. Zudem kennen sie sich mit der Analyse komplexerer Strukturen, etwa Zeitreihen oder Graphen, aus.		
Inhalte	Das Thema dieses Kurses ist Data Mining, grob übersetzbar mit "Wissensentdeckung in Datenmengen/-banken". Die Bedeutung dieses Themengebiets ist in den letzten Jahren rasant gewachsen. Die Zielsetzung besteht darin, Strukturen, Zusammenhänge sowie Gruppen ähnlicher Objekte in sehr großen Datenmengen zu erkennen und zu bewerten. Der Kurs vermittelt zunächst Kenntnisse zur Vorbereitung von Data-Mining-Methoden hinsichtlich der Charakterisierung (z.B. Klassifizierung von Attributtypen, Visualisierung) und Vorverarbeitung der Daten (etwa durch Eliminierung von Ausreißern, Aggregation oder Normalisierung). Darauf aufbauend, werden verschiedene Techniken zur Mustersuche (z.B. Apriori-Algorithmus), Klassifikation (u.a. Entscheidungsbäume, Klassifikation nach Bayes) und Clusteranalyse (beispielsweise k-Means, DBSCAN) sowie passende Evaluationsmethoden vorgestellt. Zudem erläutert der Kurs, wie komplexere Strukturen, d.h. Datenströme, Textdokumente, Zeitreihen, diskrete Folgen, Graphen sowie Webdaten, analysiert werden können. Ein Kapitel mit praktischen Beispielen in Weka bildet den Abschluss des Kurses.		
Inhaltliche Voraussetzung	Keine		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum		
Anmerkung			
Formale Voraussetzung	keine		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematisch-technische Softwareentwicklung M.Sc. Informatik M.Sc. Praktische Informatik		



## Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete Prüfungsklausur

keine

Stellenwert  
der Note 1/12

Lehrende/r	Uta Störl	Modulbeauftragte/r	Uta Störl
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01614	Architektur und Implementierung von Datenbanksystemen	SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 160 Stunden Bearbeitung der Einsendeaufgaben inkl. Verarbeitung des Korrektur-Feedbacks: 80 Stunden Wiederholung und Prüfungsvorbereitung, Prüfung: 60 Stunden		
Qualifikationsziele	Studierende erwerben in diesem Modul einen guten Überblick wie auch Detailkenntnisse der Architektur und Implementierung von Datenbanksystemen. Sie können die Schichtenarchitektur und die Aufgaben der jeweiligen Komponenten der Implementierung erläutern. Sie können verschiedene Indexstrukturen im Detail erklären. Die Schritte der Verarbeitung einer SQL-Anfrage in der Optimierung und Planerzeugung können von ihnen beschrieben werden. Sie können erklären, wann verzahnte Abläufe von Transaktionen als korrekt anzusehen sind und wie der Transaktionsmanager des Datenbanksystems sicherstellt, dass nur solche Abläufe zugelassen werden. Sie können erklären, wie nach Systemabstürzen der korrekte Zustand der Datenbank wiederhergestellt wird. Die wichtigsten Konzepte verteilter Datenbankarchitekturen und insbesondere die Unterschiede zu nicht-verteilten Architekturen können von den Studierenden erläutert werden. Basierend auf diesen Kenntnissen sind sie in der Lage, Effizienzproblemen in Datenbankarchitekturen auf den Grund zu gehen.		
Inhalte	<ul style="list-style-type: none"> <li>- Referenzarchitekturen für die Implementierung von Datenbanksystemen</li> <li>- Externspeicher- und Systempufferverwaltung</li> <li>- Indexstrukturen</li> <li>- Anfrageverarbeitung und Optimierung, insbesondere die Ermittlung eines kostengünstigen Plans für eine gegebene SQL-Anfrage</li> <li>- Transaktionsmanagement im Mehrbenutzerbetrieb</li> <li>- Ausfallsicherheit und Wiederherstellung nach Fehlern von Soft- und Hardware (Recovery)</li> <li>- Verteilte Datenbankarchitekturen</li> </ul>		
Inhaltliche Voraussetzung	Kenntnisse der Konzepte von Standard-Datenbanksystemen, z.B. aus dem Modul 63012 Softwaresysteme (01671 Datenbanken I), 63017 Datenbanken und Sicherheit im Internet (01671 Datenbanken I) oder 63118 Datenbanken I (01671).		
Lehr- und Betreuungsformen	Kursmaterial Betreuung und Beratung durch Lehrende Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Video-Meetings		
Anmerkung	Nicht zusammen mit dem Kurs „Implementierungskonzepte von Datenbanksystemen“ (01664) nutzbar		
Formale Voraussetzung	keine		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematisch-technische Softwareentwicklung		

B.Sc. Wirtschaftsinformatik  
M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

### Prüfungsformen

Prüfung  
Stellenwert 1/12  
der Note

### Art der Prüfungsleistung

bestandene benotete mündliche  
Modulprüfung

### Voraussetzung

keine

Lehrende/r	Jörg M. Haake	Modulbeauftragte/r	Jörg M. Haake
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01678 Verteilte Systeme		WS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 20 Std.): 140 Stunden Bearbeitung der Einsendeaufgaben inkl. Verarbeitung des Korrektur-Feedbacks (7 mal 10 Std.): 70 Stunden Mitwirkung an den Diskussionen in der Kurs-Newsgroup: 20 Stunden Wiederholung und Prüfungsvorbereitung: 70 Stunden		
Qualifikationsziele	Die Teilnehmenden entwickeln ein grundlegendes Verständnis für Design und Implementierung von verteilten Systemen auf der Basis moderner Betriebssysteme und Rechnernetze. Sie können gängige Probleme bei verteilten Systemen mittels Designprinzipien lösen und die Einsatzmöglichkeiten und Realisierungsmöglichkeiten verteilter Anwendungen beurteilen. Durch die Teilnahme an der Kurs-Newsgroup wird das Einüben wissenschaftlicher Kommunikation gefördert.		
Inhalte	Das Modul behandelt die Funktionsweise und Designprinzipien von verteilten Systemen, die zum Verständnis heutiger Anwendungssysteme im Internet unentbehrlich sind. Ein verteiltes System besteht aus mehreren Komponenten, die auf vernetzten Rechnern installiert sind und ihre Aktionen durch den Austausch von Nachrichten über Kommunikationskanäle koordinieren. Im Vergleich zu autonomen Rechensystemen treten bei verteilten Systemen ganz neue Probleme auf: Daten, welche auf unterschiedlichen Rechensystemen auch unterschiedlich dargestellt werden, sollen ausgetauscht werden, Prozesse müssen synchronisiert werden, verteilte persistente Datenbestände sollen konsistent gehalten werden. Schwerpunktmäßig behandelt werden die Grundlagen verteilter Systeme, Prozesse und Kommunikation, Namen und Synchronisierung, Konsistenz und Replikation, Fehlertoleranz, Sicherheit und verteilte Dateisysteme.		
Inhaltliche Voraussetzung	Inhalte der Modulr 63811 "Einführung in die imperative Programmierung" (01613) und 63113 "Datenstrukturen" (01663) und des Kurses 01801 "Betriebssysteme und Rechnernetze" aus dem Modul 63012 "Softwaresysteme" oder vergleichbare Kenntnisse sowie Erfahrungen im Umgang mit einem verbreiteten Betriebssystem wie Unix, MacOS oder Windows. Wenn Ihnen Grundkenntnisse über Betriebssysteme oder Rechnernetze fehlen, so sollte es für Sie möglich sein, ergänzend zur Bearbeitung des Kurses die Ihnen unbekannt Sachverhalte in einschlägigen Fachbüchern nachzulesen. Eine gewisse Erfahrung im Programmieren mit einer Programmiersprache wie Java oder C sollten Sie auch mitbringen, um einige Beispiele zu verstehen.		
Lehr- und Betreuungsformen	Kursmaterial internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende Einsendeaufgaben mit Korrektur und/oder Musterlösung Studententag/e		
Anmerkung	Ab dem WS 2019/20 wird das Modul Verteilte Systeme mit dem Kurs 01678 Verteilte Systeme im Wintersemester in Form einer zweistündigen Klausurarbeit und im Sommersemester in Form einer mündlichen Prüfung geprüft.  Der Basistext muss vor Semesterbeginn beschafft werden. Basistext:  Andrew S. Tanenbaum, Maarten van Steen: Distributed Systems: Principles and		

Paradigms, Prentice Hall 2006, ISBN 0-13-239227-5,  
oder 2008, ISBN 0-13-613553-6 (Paperback)  
oder Andrew S. Tanenbaum, Maarten van Steen: Distributed Systems: Principles and  
Paradigms, Prentice Hall; Auflage: new edition. (26. Juli 2013) ISBN-13:  
978-1292025520 (Taschenbuch)

oder die deutsche Übersetzung:

Andrew S. Tanenbaum, Maarten van Steen: Verteilte Systeme: Prinzipien und  
Paradigmen, Pearson Studium 2007, ISBN 3-8273-7293-3

Bitte verwenden Sie nur die zweite Auflage des Buches von Tanenbaum/van Steen!

Formale Voraussetzung keine

Verwendung des Moduls B.Sc. Informatik  
B.Sc. Mathematisch-technische Softwareentwicklung  
B.Sc. Wirtschaftsinformatik  
M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

<b>Prüfungsformen</b>	Art der Prüfungsleistung	Voraussetzung
Prüfung	s. Anmerkung	keine
Stellenwert der Note	1/12	

Lehrende/r	Gabriele Peters	Modulbeauftragte/r	Gabriele Peters
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01697 Einführung in Mensch-Computer-Interaktion		WS/SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 150 Stunden Bearbeiten der Selbsttest- und Einsendeaufgaben: 75 Stunden Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	Durch die Teilnahme an diesem Kurs erhalten die Studierenden einen Überblick über Entwicklungen, Begriffe und Zusammenhänge im Kontext der Mensch-Computer-Interaktion. Darüber hinaus sind sie mit den Grundlagen der menschlichen Wahrnehmung vertraut. Die Studierenden lernen die wesentlichen Entwicklungen und Zusammenhänge im Umfeld der Mensch-Computer-Interaktion kennen. Sie besitzen das Wissen, wie man die Eigenschaften der menschlichen Wahrnehmung gezielt bei der Gestaltung interaktiver Systeme berücksichtigen kann und sie kennen verschiedene Methoden und Verfahren, die es erlauben, die Entwicklung und Evaluation interaktiver Systeme durchzuführen und in den regulären Softwareentwicklungsprozess zu integrieren.		
Inhalte	Der Kurs führt in die grundlegenden Konzepte und Techniken des Gebiets Mensch-Computer-Interaktion (MCI) ein. Er beginnt mit einer Übersicht über die bisherige Entwicklung dieses Teilgebiets der Informatik sowie einer Klärung und Definition seiner Grundbegriffe. Im Anschluss werden die möglichen technischen Schnittstellen einer Interaktion zwischen Mensch und Computer (haptische, auditive und visuelle) beschrieben und hinsichtlich ihrer charakteristischen Eigenschaften untersucht. Dieser Betrachtung der technischen Seite der MCI folgt eine Einführung in die neurobiologischen Grundlagen der menschlichen Wahrnehmung am Beispiel der visuellen Informationsverarbeitung. Ausgewählte wahrnehmungspsychologische Phänomene werden beschrieben, aus denen sich schließlich grundlegende Prinzipien für die Gestaltung von Interaktion herleiten lassen. Anschließend werden verschiedene Aspekte der Implementierung interaktiver Systeme beleuchtet. Die abschließende Kurseinheit befasst sich mit der Evaluation von Funktionalität und Bedienbarkeit von Benutzungsschnittstellen mithilfe statistischer Methoden.		
Inhaltliche Voraussetzung	Mathematik-Kenntnisse, die durch die Erlangung der allgemeinen Hochschulreife erworben wurden		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Zusatzmaterial Betreuung und Beratung durch Lehrende		
Anmerkung	-		
Formale Voraussetzung	keine		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematisch-technische Softwareentwicklung B.Sc. Wirtschaftsinformatik		

M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

## Prüfungsformen

Prüfung

Art der Prüfungsleistung

bestandene benotete Prüfungsklausur

Voraussetzung

Als Zulassungsvoraussetzung für die Modulprüfungsklausur müssen bei 6 von 7 Kurseinheiten 75% der Einsendeaufgaben erfolgreich bearbeitet werden. Bei einer weiteren Kurseinheit reicht es aus, mehr als 50% der Punkte zu erzielen.

Stellenwert  
der Note 1/12

Lehrende/r	Gabriele Peters	Modulbeauftragte/r	Gabriele Peters
	Dauer des Moduls ein oder zwei Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01698	Interaktive Systeme I: Konzepte und Methoden des Computersehens	WS/SS SWS 2+1
	01699	Interaktive Systeme II: Konzepte und Methoden bildbasierter 3D-Rekonstruktion	WS/SS SWS 2+1
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 150 Stunden Bearbeiten der Selbsttest- und Einsendeaufgaben: 75 Stunden Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	In beiden Kursen erlangen die Studierenden fundierte Kenntnisse sowohl der theoretischen, mathematischen Grundlagen als auch der vorgestellten, anwendungsorientierten Konzepte und Methoden. Die Studierenden besitzen ein solides Wissen über digitale Signalverarbeitungsmethoden, z.B. die Fouriertransformation und die derzeit wichtigsten Verfahren der Bildverarbeitung. Darüber hinaus kennen die Studierenden weiterführende Datenverarbeitungsmethoden wie z.B. Clusteringverfahren oder die Verwendung probabilistischer Modelle. Desweiteren besitzen die Studierenden Kenntnisse über Methoden der dreidimensionalen Bildrekonstruktion sowie Verfahren der linearen und nicht-linearen Optimierung.		
Inhalte	Der Kurs 01698 führt zunächst in Konzepte und Methoden der allgemeinen Signalverarbeitung und -interpretation ein. Darauf aufbauend werden wesentliche Konzepte und Methoden des Computersehens und weiterführender Signalverarbeitungskonzepte vermittelt. Es werden u.a. die Eigenschaften linearer Systeme, die Fouriertransformation, Methoden des Computersehens, sowie Clusteringverfahren und modellbasierte Methoden der Signalinterpretation im Detail behandelt. Der Kurs 01699 behandelt Konzepte und Methoden, mit deren Hilfe sich eine dreidimensionale, visuelle Darstellung eines realen Objektes aus einer Reihe von zweidimensionalen Bildern errechnen lässt. Hierzu erfolgt zunächst eine Einführung in die mathematischen Grundlagen wie etwa die projektive Geometrie. Anschließend werden Methoden vorgestellt, die es ermöglichen, eine 3D-Punktwolke aus einer Reihe von 2D-Bildern zu errechnen und anschließend zu triangulieren.		
Inhaltliche Voraussetzung	Mathematik-Kenntnisse, die den im Modul 61111 "Mathematische Grundlagen" (01141) vermittelten Kenntnissen entsprechen.		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende		
Anmerkung	-		
Formale Voraussetzung	keine		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematisch-technische Softwareentwicklung B.Sc. Wirtschaftsinformatik		



M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

## Prüfungsformen

Prüfung

Art der Prüfungsleistung

bestandene benotete Prüfungsklausur

Voraussetzung

Als Zulassungsvoraussetzung für die Klausur müssen in beiden Kursen bei jeweils 3 von 4 Kurseinheiten 75% der Einsendeaufgaben erfolgreich bearbeitet werden. Bei jeweils einer weiteren Kurseinheit reicht es aus, mehr als 50% der Punkte zu erzielen.

Stellenwert  
der Note 1/12

63514

## Simulation

Lehrende/r	Jörg Keller	Modulbeauftragte/r	Jörg Keller
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01610 Simulation		SS SWS 3+3
Detaillierter Zeitaufwand	Kurseinheiten: 100 Stunden Einsendearbeiten: 150 Stunden Prüfungsvorbereitung: 50 Stunden		
Qualifikationsziele	Studierende kennen die Prinzipien der Mikro- und Makrosimulation und können Anwendungsszenarien in Simulationen übersetzen sowie Simulationsergebnisse hinsichtlich ihrer Bedeutung in der Anwendung interpretieren.		
Inhalte	<p>Vereinfachte Ausschnitte der Wirklichkeit und rechnergestützte Simulationen dieser Ausschnitte finden sich in vielen Anwendungsbereichen der Informatik. Gleichzeitig liegt der Simulation eine einheitliche Methodik zugrunde, die in der anwendungsgetriebenen Nutzung aber nur wenig aufscheint und daher in einem eigenen Modul vermittelt wird.</p> <p>Nach einer Einführung in die Grundlagen von Simulation, Spieltheorie und Scheduling werden Simulationen auf makro- und mikroskopischer Ebene sowie mit stochastischen Ansätzen für den Bereich des Verkehrs gezeigt. Aus dem Bereich der Physik werden Simulationen für Molekularbewegung behandelt. Ein Ausflug in die Welt der Populationen und des Chaos schließt das Modul ab.</p>		
Inhaltliche Voraussetzung	Inhaltliche Voraussetzungen: Module 61111 "Mathematische Grundlagen" (01141), 63811 "Einführung in die imperative Programmierung" (01613), 63113 "Datenstrukturen und Algorithmen" (01663) und 63081 "Grundpraktikum Programmierung" (01584)		
Lehr- und Betreuungsformen	<p>Kursmaterial</p> <p>Einsendeaufgaben mit Korrektur und/oder Musterlösung</p> <p>internetgestütztes Diskussionsforum</p> <p>Betreuung und Beratung durch Lehrende</p>		
Anmerkung	Der Basistext muss vor Semesterbeginn beschafft werden: H.-J. Bungartz: Modellbildung und Simulation, 2. Auflage, Springer Spektrum, 2013		
Formale Voraussetzung	keine		
Verwendung des Moduls	<p>B.Sc. Informatik</p> <p>B.Sc. Mathematisch-technische Softwareentwicklung</p> <p>M.Sc. Informatik</p> <p>M.Sc. Praktische Informatik</p>		
<b>Prüfungsformen</b>	Art der Prüfungsleistung	Voraussetzung	
Prüfung	bestandene benotete mündliche	keine	
Stellenwert der Note	1/12 Modulprüfung		

Lehrende/r	Wolfram Schiffmann	Modulbeauftragte/r	Wolfram Schiffmann
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01706 Anwendungsorientierte Mikroprozessoren		WS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Einüben des Stoffes, insbesondere durch Einsendeaufgaben (7 mal 15 Stunden): 105 Stunden Wiederholung und Prüfungsvorbereitung (Selbststudium): 55 Stunden		
Qualifikationsziele	Nach dem Bearbeiten des Kurses verstehen die Studierenden den komplexen Aufbau anwendungsorientierter Mikroprozessoren und das Zusammenwirken ihrer Komponenten. Außerdem wissen sie, wie ein einfacher Mikroprozessor in seine analoge oder digitale "Umwelt" eingebettet ist und mit ihr kommuniziert. Dadurch werden Ausbildungslücken geschlossen, die in vielen Kursen über Mikroprozessortechnik bleiben, die sich hauptsächlich mit den "High-End"-Prozessoren und ihren komplexen Komponenten beschäftigen. Nach dem Bearbeiten des Kurses sind die Studierenden in der Lage, den Einsatz z.T. sehr einfacher Mikroprozessoren in den Hunderten von technischen Geräten (Fernbedienungen, Mobiltelefone, Haushaltsgeräte usw.) zu verstehen, die ihnen täglich das Leben erleichtern.		
Inhalte	Der Kurs beschäftigt sich mit der Architektur und der Funktionsweise von anwendungsorientierten Mikroprozessoren. Das sind zum einen die Mikrocontroller, die im Prinzip vollständige Rechner in einem einzigen Baustein darstellen, zum anderen die auf die Verarbeitung digitalisierter analoger Signale spezialisierten Digitalen Signalprozessoren. Im Mittelpunkt des Kurses stehen technisches Grundlagenwissen und praktischer Einsatz. Es wird gezeigt, aus welchen Komponenten diese Mikroprozessoren aufgebaut sind und wie diese zusammenwirken. Dabei wird insbesondere hervorgehoben, wie sie an ihre spezifischen Anwendungen angepasst sind. Als Grundlage für die Programmierung der Prozessoren wird ihre Schnittstelle zwischen der Hardware und Software ausführlich behandelt. Für beide Prozessortypen werden Produktbeispiele präsentiert. Ein weiterer Schwerpunkt wird auf die Beschreibung der Komponenten gelegt, die einen Mikroprozessor zu einem Mikrocontroller erweitern, also insbesondere die verschiedenen Speicherbausteine, Bussysteme sowie Schnittstellen- und Systemsteuerbausteine.		
	<b>Ergänzende Literatur:</b> H. Bähring: Mikrorechner-Technik, 2 Bände, Springer Verlag, 2002, ISBN: 3-540-41648 X, 3-540-43693-6 W. Schiffmann: Technische Informatik 2, Springer Verlag, 2002, ISBN: 3-540-43854-8 U. Brinkschulte, T. Ungerer: Mikrocontroller und Mikroprozessoren, Springer Verlag, 2007, ISBN: 978-3-540-46801-1 H. Bähring: Anwendungsorientierte Mikroprozessoren, Springer Verlag, 2010, ISBN: 978-3-642-12291-0		
Inhaltliche Voraussetzung	Grundkenntnisse in Digitaltechnik und elektrotechnischen Grundlagen		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende		
Anmerkung	-		
Modulhandbuch	M.Sc. Informatik		

Formale Voraussetzung keine

Verwendung des Moduls B.Sc. Informatik  
B.Sc. Mathematisch-technische Softwareentwicklung  
M.Sc. Informatik  
M.Sc. Praktische Informatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/12	Modulprüfung	

Lehrende/r	Lena Oden	Modulbeauftragte/r	Lena Oden
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01727	Parallele Programmierung und Grid-Computing	
		WS	SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 150 Stunden Bearbeiten der Einsendearbeiten: 75 Stunden Studientage und Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	Nachdem die Studierenden das Modul bearbeitet haben, können sie bei der Lösung komplexer Problemstellungen parallelisierbare Komponenten identifizieren, auf homogene oder heterogene Prozessorarchitekturen verteilen, Softwareimplementierungen für diese Rechnerarchitekturen konstruieren, Testfälle generieren und damit die parallele Implementierung evaluieren, Fehler in einer Implementierung identifizieren und beheben, Optimierungsmöglichkeiten gegenüberstellen und beurteilen, die Implementierung rekonstruieren und somit möglichst gut angepasste parallele Softwareimplementierungen für die einzelnen Problemstellungen hervorbringen.		
Inhalte	Mit dem Aufkommen von Multicore-Prozessoren in Desktop-PCs verlässt die parallele Programmierung die Nischenecke der Großrechner und wird für eine Vielzahl von Anwendungen interessant. Gleichzeitig werden traditionelle Arbeitsfelder von Parallelrechnern zunehmend durch das Grid-Computing erobert. Der Kurs enthält Beiträge zu folgenden Themengebieten: Grundlagen und Modelle der parallelen Programmierung, Parallele Programmieretechniken wie Shared Memory Programmierung mit POSIX Threads, Message Passing Interface (MPI) und OpenMP, parallele Matrizenrechnung, parallele Graphalgorithmen, Einführung in das Cluster- und Grid-Computing, Einführung in die Middleware Condor, Scheduling von Metatasks, Fallstudien realer Grid-Systeme und grundlegende Scheduling-Techniken für Workflows in Grids sowie eine kurze Einführung in Virtuelle Maschinen und Cloud-Computing. Für die Übungen werden verschiedene parallele Computersysteme bereitgestellt und die Studierenden müssen selbst parallele Software erstellen.		
	<b>Ergänzende Literatur:</b> B.Wilkinson, M. Allen: Parallel Programming, Second Edition, Pearson Education International, 2005, ISBN 0-13-191865-6 A. Gramma, A. Gupta, G. Karypis, V. Kumar: Introduction to Parallel Computing, Second Edition, Addison Wesley, 2003, ISBN 0-201-64865-2 B. Jacob Elektrotechnik al.: Introduction to Grid Computing, IBM Redbook, <a href="http://ibm.com/redbooks">http://ibm.com/redbooks</a> Barry Wilkinson: Grid Computing, Chapman & Hall, 2009		
Inhaltliche Voraussetzung	Kenntnisse aus Modul 63013 "Computersysteme" (01608/01609), Modul 63811 "Einführung in die imperative Programmierung" (01613), Modul 61113 "Datenstrukturen und Algorithmen" (01663), Modul 64311 "Betriebssysteme und Rechnernetze" (01690) sowie Modul 63114 "Datenbanken I" (01666)		
Lehr- und Betreuungsformen	Kursmaterial internetgestütztes Diskussionsforum Zusatzmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung Betreuung und Beratung durch Lehrende		
Anmerkung	-		
Formale Voraussetzung	keine		

Verwendung des Moduls B.Sc. Informatik  
B.Sc. Mathematisch-technische Softwareentwicklung  
B.Sc. Wirtschaftsinformatik  
M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/12	Modulprüfung	

Lehrende/r	Lars Mönch	Modulbeauftragte/r	Lars Mönch
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01770	Betriebliche Informationssysteme	WS/SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 150 Stunden, Bearbeiten der Übungsaufgaben: 75 Stunden, Wiederholung des Stoffs, Studientage und Prüfungsvorbereitung: 75 Stunden.		
Qualifikationsziele	Die Studierenden kennen ein Gesamtkonzept der integrierten betrieblichen Informationsverarbeitung. Die Studierenden werden mit dem Architekturbegriff für betriebliche Informationssysteme vertraut gemacht und kennen ausgewählte Architekturkonzepte. Sie werden mit der Konstruktion betrieblicher Informationssysteme vertraut gemacht. Die Studierenden werden in die Lage versetzt, eigenständig Auswahlentscheidungen für betriebswirtschaftliche Standardsoftware treffen zu können. Die Studierenden werden mit grundlegenden Funktionen und Prozessen im Produktionssektor und im Vertrieb eines Industriebetriebs vertraut gemacht. Weiterhin werden den Studierenden Kenntnisse über die Architektur und die Funktionsweise ausgewählter Informationssysteme für den Produktions- und Vertriebssektor vermittelt.		
Inhalte	Dieser Kurs stellt Grundlagen, Konzepte und Techniken des Gebiets "Betriebliche Informationssysteme" bereit und behandelt die Themen Integrierte Informationsverarbeitung, Architekturen betrieblicher Informationssysteme, Konstruktion betrieblicher Informationssysteme, Anwendungssysteme, Funktionen und Prozesse im Produktions- und Vertriebssektor. Außerdem werden an ausgewählten Beispielen für betriebliche Informationssysteme die genannten Themen exemplarisch vertieft.		
	Betriebliche Anwendungssoftware hat sich in den letzten Jahren von monolithischen Systemen hin zu komponentenbasierten, dienstorientierten Softwaresystemen entwickelt. Moderne unternehmensweite Software besteht aus Komponenten zur Lösung betrieblicher Problemstellungen und aus Komponenten, die unabhängig von den betrieblichen Aufgaben sind und zum Beispiel Vermittlungsfunktionalität, Datenhaltung, Ablauflogik sowie das Betriebssystem zur Verfügung stellen. Es wird gezeigt, wie moderne Technologien wie Middleware, XML und Webservices für die Implementierung von betrieblichen Informationssystemen verwendet werden.		
Inhaltliche Voraussetzung	Absolvierung des 1. und 2. Semesters ("Einführung in die Wilnf. und Modellierung von Informationssystemen"), Modul 63611 "Einführung in die objektorientierte Programmierung" (01618) und 63012 "Softwaresysteme" (01671), Grundkenntnisse in BWL, insbesondere über die Funktionsweise eines Unternehmens, sind für das Verständnis des Stoffes nützlich.		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum		
Anmerkung	-		
Formale Voraussetzung	keine		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematisch-technische Softwareentwicklung		

B.Sc. Wirtschaftsinformatik  
M.Sc. Informatik  
M.Sc. Praktische Informatik

## Prüfungsformen

Prüfung

Art der Prüfungsleistung

bestandene benotete Prüfungsklausur

Voraussetzung

Eine Zulassung zur Klausur erfolgt, wenn insgesamt mindestens 50 % der möglichen Punkte der Einsendeaufgaben in zwei vom Lehrgebiet festgelegten Einsendeaufgaben erreicht wurden.

Stellenwert  
der Note 1/12



Lehrende/r	Lars Mönch	Modulbeauftragte/r	Lars Mönch
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01771	Entscheidungsmethoden in unternehmensweiten Softwaresystemen	SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 150 Stunden, Bearbeiten der Übungsaufgaben: 75 Stunden, Wiederholung des Stoffs und Prüfungsvorbereitung: 75 Stunden.		
Qualifikationsziele	Die Studierenden kennen die Anwendungsmöglichkeiten und -grenzen von diskreter Simulation zur Entscheidungsunterstützung in PPS- und SCM-Systemen. Die Studierenden werden insbesondere mit der grundsätzlichen Wirkungsweise diskreter Simulationssoftware vertraut gemacht. Die Studierenden werden vertieft mit den Modellierungsmethoden für Produktionssysteme vertraut gemacht. Insbesondere werden die Studierenden in die Lage versetzt, Modellierungs- und Simulationstätigkeiten für Produktionssysteme eigenständig auszuführen. Die Studierenden erwerben Kenntnisse bezüglich der Verifikation und der Validierung von Simulationsmodellen. Die Studierenden werden mit ausgewählten Planungsproblemen sowie Entscheidungsmodellen und -methoden in den Bereichen Ablaufplanung sowie Lieferkettenmanagement vertraut gemacht und können wichtige Techniken der Modellierung derartiger Probleme in APS-Systemen anwenden.		
Inhalte	Dieser Kurs behandelt ausgewählte Entscheidungsmodelle und -methoden, die in unternehmensweiten Softwaresystemen Anwendung finden. Im Vordergrund stehen dabei die diskrete ereignisorientierte Simulation und Entscheidungsmodelle und -methoden in APS- und SCM-Systemen. Typische Betrachtungsgegenstände der Modellierung und Simulation mit dem Fokus auf diskreter Simulation für Produktionssysteme werden behandelt. Die einzelnen Schritte einer Simulationsstudie werden beschrieben. Der Kurs behandelt die Funktionsweise moderner diskreter Simulationssoftware. Typische Betrachtungsgegenstände der Modellierung und Simulation von Produktionssystemen werden eingeführt. Weiterer Gegenstand des Kurses sind Planungs- und -steuerungsprobleme für die Produktionsdomäne.		
Inhaltliche Voraussetzung	Kenntnisse der Inhalte der Module 61411 "Algorithmische Mathematik" (01142) und 64111 "Betriebliche Informationssysteme" (01770)		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum		
Anmerkung	-		
Formale Voraussetzung	keine		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematisch-technische Softwareentwicklung B.Sc. Wirtschaftsinformatik M.Sc. Informatik M.Sc. Praktische Informatik M.Sc. Wirtschaftsinformatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

bestandene benotete mündliche  
Modulprüfung

Voraussetzung

Eine Zulassung zur Prüfung erfolgt, wenn insgesamt mindestens 50 % der möglichen Punkte der Einsendeaufgaben in zwei vom Lehrgebiet festgelegten Einsendeaufgaben erreicht wurden.

Lehrende/r	Christoph Beierle	Modulbeauftragte/r	Christoph Beierle
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01696 Wissensbasierte Systeme		SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 130 - 150 Stunden, Bearbeiten der Übungsaufgaben: 60 - 75 Stunden, Studientage und Prüfungsvorbereitung: 60 - 75 Stunden		
Qualifikationsziele	Die Studierenden können grundlegende Kenntnisse der wichtigsten Formalismen und Techniken der Wissensrepräsentation und Inferenz sowie Verständnis für deren sinnvollen Einsatz in realen Systemen demonstrieren. Sie können zentrale Verfahren wissensbasierter Systeme auf entsprechende Problemstellungen anwenden. Dazu zählen Repräsentation von einfachen Sachverhalten mit formaler Logik, Inferenzen in regelbasierten Systemen, Lernen von Entscheidungsbäumen und von Konzepten, Datamining mit dem Apriori-Verfahren.		
Inhalte	<p>Wissensbasierte Systeme unterscheiden sich von herkömmlichen Softwaresystemen dadurch, dass in ihnen bereichsspezifisches Wissen in einer mehr oder weniger direkten Form repräsentiert ist und zur Anwendung kommt. Typische Beispiele für wissensbasierte Systeme sind Expertensysteme, die das Fachwissen und die Schlussfolgerungsfähigkeit von Experten nachbilden. Für wissensbasierte Systeme werden daher komplexe Instrumente zur maschinellen Repräsentation, Verarbeitung und Nutzung von Wissen benötigt. Für die vielfältigen Anwendungsmöglichkeiten steht ein reichhaltiges Repertoire an Methoden der Wissensrepräsentation und der Inferenz zur Verfügung. Der Kurs soll grundlegende Kenntnisse der wichtigsten Formalismen und Techniken vermitteln, darüber hinaus aber auch ein Verständnis für deren sinnvollen Einsatz in realen Systemen. So veranschaulicht eine Vielzahl praktischer Beispiele Möglichkeiten und Grenzen wissensbasierter Systeme.</p> <p>Die Themenbereiche des Kurses sind im Einzelnen: Aufbau und Arbeitsweise wissensbasierter Systeme, logikbasierte Wissensrepräsentation und Inferenz, regelbasierte Systeme, maschinelles Lernen, Data Mining und Wissensfindung in Daten, fallbasiertes Schließen, Problemstellungen bei der Verwendung nichtmonotonen Schließens und quantitativer Methoden.</p> <p><b>Ergänzende Literatur:</b>  C. Beierle, G. Kern-Isberner. Methoden wissensbasierter Systeme - Grundlagen, Algorithmen, Anwendungen. Springer Vieweg, 6. überarbeitete Auflage, 2019.  S. Russell, P. Norvig, Künstliche Intelligenz: ein moderner Ansatz, Pearson Studium, 2004</p>		
Inhaltliche Voraussetzung	-		
Lehr- und Betreuungsformen	Kursmaterial Einsendaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Studientag/e		
Anmerkung	Ab Wintersemester 2022/2023 wird das Modul regelmäßig in jedem Wintersemester angeboten.		
Formale Voraussetzung	keine		
Modulhandbuch	M.Sc. Informatik		

Verwendung des Moduls B.Sc. Informatik  
B.Sc. Mathematisch-technische Softwareentwicklung  
B.Sc. Wirtschaftsinformatik  
M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete Prüfungsklausur	keine
Stellenwert der Note	1/12		

Lehrende/r	Mario Kubek	Modulbeauftragte/r	Mario Kubek
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01864 Mobile Security		WS/SS SWS 4+2
Detaillierter Zeitaufwand	Kurseinheiten: 150 Stunden Übungsaufgaben: 75 Stunden Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	Die Studierenden haben nach erfolgreicher Bearbeitung fundierte Kenntnisse zu den jeweiligen Sicherheitsarchitekturen und -mechanismen moderner, mobiler Betriebssysteme wie iOS und Android erlangt. Sie kennen typische Bedrohungen, Angriffsszenarien und Gegenmaßnahmen im Kontext mobiler Geräte, Applikationen und Datenübertragung. Die Studierenden sind zudem in der Lage, selbstständig mobile Applikationen auf Sicherheitsprobleme und Schadcode hin zu analysieren und sind mit dem dafür nötigen Vorgehen und gängigen Werkzeugen vertraut. Durch dieses Wissen können die Studierenden den Sicherheitsstatus ihrer Endgeräte und der darauf installierten Applikationen einschätzen und ihn selbst aktiv verbessern.		
Inhalte	Der Kurs "Mobile Security" führt in die Sicherheitskonzepte und -mechanismen mobiler Endgeräte wie Smartphones und Tablets sowie der auf ihnen laufenden Betriebssysteme und Applikationen ein. Der Fokus dieser Betrachtungen liegt dabei auf den gängigen Betriebssystemen iOS und insbesondere Android. Konkret befasst sich der Kurs zunächst mit den allgemeinen Bedrohungen und Angriffsszenarien in diesem Kontext sowie den Sicherheitsarchitekturen obiger Plattformen und ihren Prinzipien als Gegenmaßnahmen. Der zweite Schwerpunkt ist den Sicherheitsproblemen und der Einführung in das Penetration Testing mobiler Applikationen gewidmet. Die dazu nötigen Techniken der statischen und dynamischen Analyse werden vorgestellt und voneinander abgegrenzt. In diesem Rahmen wird die Vorgehensweise beim Reversing von Android-Applikationen erklärt, wobei zu diesem Zweck auf ihre Beschaffung, ihre Analyse und die dafür nötigen technischen Umgebungen und Werkzeuge eingegangen wird. Weiterhin werden die wichtigsten Schwachstellen im Code mobiler Applikationen und deren Erkennung sowie die Detektion von Schadcode und gängige Schutzmaßnahmen behandelt. Ebenfalls werden verschiedene Ansätze forensischer Untersuchungen mobiler Endgeräte besprochen. Abschließend gibt der Kurs einen Überblick über eine Reihe von Angriffen auf die Datenübertragung und das dafür nötige Vorgehen.		
Inhaltliche Voraussetzung	Modul 63512 "Sicherheit im Internet" (01866/01868)		
Lehr- und Betreuungsformen	Kursmaterial Einsendaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Zusatzmaterial Betreuung und Beratung durch Lehrende		
Anmerkung	Der Basistext muss vor Semesterbeginn beschafft werden. Basistext: M. Spreitzenbarth: Mobile Hacking: Ein kompakter Einstieg ins Penetration Testing mobiler Applikationen - iOS, Android und Windows Phone, 2017		
Formale Voraussetzung	keine		

Verwendung des Moduls B.Sc. Informatik  
B.Sc. Mathematisch-technische Softwareentwicklung  
M.Sc. Informatik  
M.Sc. Praktische Informatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete Prüfungsklausur	keine
Stellenwert der Note	1/12		

# Katalog M1: Formale Methoden und Algorithmik

Lehrende/r	Silke Hartlieb	Modulbeauftragte/r	Silke Hartlieb
Dauer des Moduls ein Semester		ECTS	10
Workload		Häufigkeit	in jedem Wintersemester
		300 Stunden	
Lehrveranstaltung(en)	01321 Mathematische Grundlagen der Kryptografie	WS	SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 25 Stunden): 175 Stunden Einüben des Stoffes (z.B. durch Einsendeaufgaben): 75 Stunden Wiederholung und Prüfungsvorbereitung (u.a. Studientag): 50 Stunden		
Qualifikationsziele	Die Studierenden lernen klassische und aktuelle Verfahren der Kryptografie kennen und verstehen die mathematischen Hintergründe dieser Verfahren. Sie kennen die für den Bereich IT-Sicherheit wichtigsten Inhalte der Algebra und Elementaren Zahlentheorie und wissen, wie diese mathematischen Grundlagen in das Design von Kryptosystemen und in die Kryptoanalyse einfließen.		
Inhalte	Die Kryptografie ist die Lehre von den Geheimschriften. Während diese bis vor wenigen Jahren eine Domäne des Militärs und der Diplomatie war, hält sie nun im Zuge der elektronischen Datenverarbeitung und Kommunikation mehr und mehr Einzug ins tägliche Leben. Neben der Aufgabe, Inhalte von Nachrichten vor der Nutzung von Unbefugten zu schützen, sind noch andere Aufgaben hinzugekommen, wie etwa sicherzustellen, dass eine Nachricht im Zuge der Übermittlung nicht geändert wurde, oder dass sie wirklich von dem angegebenen Absender stammt. In dem Kurs werden zunächst klassische symmetrische Verfahren der Kryptografie vorgestellt. Im Zentrum stehen jedoch Public Key Verfahren, die hauptsächlich auf algebraischen und zahlentheoretischen Grundlagen basieren. Zu nennen sind elementare Gruppen- und Ringtheorie, Theorie endlicher Körper, Theorie ganzzahliger Gitter sowie modulare Arithmetik, Theorie elliptischer Kurven und Primzahltests. Diese Grundlagen werden bereitgestellt, und es wird gezeigt, wie sie in moderne Kryptosysteme einfließen und in der Kryptoanalyse eingesetzt werden. Die genauen Inhalte sind: <ul style="list-style-type: none"> <li>- Grundlagen der Algebra (Gruppen, Ringe, (endliche) Körper, elliptische Kurven)</li> <li>- Grundlagen der Elementaren Zahlentheorie</li> <li>- Asymmetrische Kryptosysteme (RSA-, Massey-Omura-, Diffie-Hellman-, ElGamal-, Kryptosystem, Kryptosysteme über elliptischen Kurven),</li> <li>- Primzahltests</li> <li>- Komplexität</li> <li>- Gitter (Basen, LLL-Algorithmus, Knapsack-Kryptosystem)</li> </ul>		
Inhaltliche Voraussetzung	Gute Kenntnisse des Moduls 61112 "Lineare Algebra" (01143) und des Moduls 61211 "Analysis" (01144). Die geforderten Voraussetzungen gehen über das hinaus, was in einem Studium der Informatik an Mathematikkenntnissen vermittelt wird.		
Lehr- und Betreuungsformen	Kursmaterial internetgestütztes Diskussionsforum Studientag/e Einsendeaufgaben mit Korrektur und/oder Musterlösung Betreuung und Beratung durch Lehrende Zusatzmaterial		
Anmerkung	-		
Formale Voraussetzung	keine		



Verwendung des Moduls B.Sc. Mathematik  
B.Sc. Mathematisch-technische Softwareentwicklung  
M.Sc. Informatik  
M.Sc. Mathematik  
M.Sc. Praktische Informatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/12	Modulprüfung	

61414

## Effiziente Graphenalgorithmen

Lehrende/r	Winfried Hochstättler	Modulbeauftragte/r	Winfried Hochstättler
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01216	Kombinatorische Optimierung - Effiziente Graphenalgorithmen	
		WS	SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Einüben des Stoffes (insbesondere durch Einsendeaufgaben; 7 mal 15 Stunden): 105 Stunden Wiederholung und Prüfungsvorbereitung (Studientag und Selbststudium): 55 Stunden		
Qualifikationsziele	Die Studierenden kennen die Grundlagen der Graphentheorie und wesentliche Datenstrukturen zur Implementierung von Graphenalgorithmen. Sie können die Laufzeit von Algorithmen abschätzen und sind sich der Problematik P vs. NP bewusst. Sie beherrschen wesentliche Algorithmen zur Baumsuche, minimalen aufspannenden Bäumen, kürzesten Wegen, maximalen Flüssen und Matchings inklusive Laufzeitanalyse und Korrektheitsbeweisen. Sie wissen was primale, duale und primal-duale Verfahren sind.		
Inhalte	Graphen und algorithmische Graphenprobleme Durchsuchen von Graphen Minimale aufspannende Bäume und Matroide kürzeste Wege maximale Flüsse Matchings Lineare Optimierungsdualität kostenminimale Flüsse und gewichtete Matchings  Ergänzende Literatur: Schrijver: Combinatorial Optimization - Polyhedra and Efficiency, Springer 2003 Cook, Cunningham, Pulleyblank, Schrijver: Combinatorial Optimization, Barnes & Noble, Wiley, 1997 Korte, Vygen: Kombinatorische Optimierung, Springer, 2012		
Inhaltliche Voraussetzung	Modul 61111 "Mathematische Grundlagen" (01141), 61411 "Algorithmische Mathematik" (01142)		
Lehr- und Betreuungsformen	internetgestütztes Diskussionsforum Studientag/e Zusatzmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung		
Anmerkung	Der Basistext muss vor Semesterbeginn beschafft werden. Basistext: Hochstättler/Schliep: CATBox - An Interactive Course in Combinatorial Optimization, Springer 2010.		
Formale Voraussetzung	keine		
Verwendung des Moduls	M.Sc. Informatik M.Sc. Mathematik M.Sc. Praktische Informatik M.Sc. Wirtschaftsinformatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

bestandene benotete mündliche  
Modulprüfung

Voraussetzung

keine

Lehrende/r	Christian Icking	Modulbeauftragte/r	Christian Icking
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01840 Algorithmische Geometrie		SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 120 Stunden Bearbeiten von Übungs- und Einsendeaufgaben: 100 Stunden Wiederholung und Prüfungsvorbereitung (Selbststudium, freiwilliger Studientag): 80 Stunden		
Qualifikationsziele	Durch diesen Kurs lernen die Studierenden einerseits die Anwendung von Algorithmen und Datenstrukturen für die Lösung von meist anschaulichen, gut motivierten und anspruchsvollen Problemen sowie andererseits auch die konsequente, mathematisch exakte Analyse von solchen Verfahren und Strukturen.		
Inhalte	<p>Die Algorithmische Geometrie beschäftigt sich mit effizienten Lösungsverfahren für geometrische Probleme. Ihre Anwendungen liegen unter anderem in den Bereichen Logistik, Robotik, Bilderzeugung und Geoinformationssysteme. In diesem Kurs werden die Grundlagen hierfür bereitgestellt und zum Beispiel folgende Fragen beantwortet: Wie bestimmt man schnell den minimalen Abstand zwischen Punkten in der Ebene? Wie berechnet man effizient Schnitte von geometrischen Objekten? Wie bestimmt man den sichtbaren Bereich in einem Raum bzw. wo platziert man dort Überwachungssysteme? Wie trianguliert man eine ebene Punktmenge? Wie verwaltet man mehrdimensionale Punktmengen effizient? Was sind die Einzugsbereiche von Versorgungsstationen oder Läden in einem einfachen ökonomischen Modell? Wie bewegt man sich in unbekannter Umgebung, um systematisch ein Ziel zu finden? Wie findet man Approximationslösungen für schwierige geometrische Optimierungsprobleme?</p> <p>Zusätzlich zum Kurstext (350 Seiten mit 250 Abbildungen) gibt es eine Sammlung von Webseiten und Java-Applets, die dynamisches Anschauungsmaterial zum Kurs bereitstellen.</p> <p><b>Ergänzende Literatur:</b></p> <p>M. de Berg, O. Cheong, M. van Kreveld, M. Overmars: Computational Geometry: Algorithms and Applications. Springer-Verlag, third edition, 2008.</p> <p>J. O'Rourke: Computational Geometry in C. Cambridge University Press, second edition, 1998.</p> <p>F. P. Preparata, M. Ian Shamos: Computational Geometry. Springer-Verlag, corrected fifth printing, 1993.</p> <p>F. Aurenhammer, R. Klein, D.-T. Lee: Voronoi Diagrams and Delaunay Triangulations, World Scientific Publishing Company 2013.</p>		
Inhaltliche Voraussetzung	Kenntnisse in Datenstrukturen und Grundkenntnisse der Mathematik, z.B. aus den Modulen 61111 "Mathematische Grundlagen" (01141) und 61411 "Algorithmische Mathematik" (01142)		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Studientag/e Zusatzmaterial Betreuung und Beratung durch Lehrende		

Anmerkung -  
Formale Voraussetzung keine

Verwendung des Moduls M.Sc. Informatik  
M.Sc. Praktische Informatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/12	Modulprüfung	

63914

## Komplexitätstheorie

Lehrende/r	André Schulz	Modulbeauftragte/r	André Schulz
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01686 Komplexitätstheorie		WS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten von Basistext und Leittext: 200 Stunden Bearbeiten von Übungs- und Einsendeaufgaben: 50 Stunden Studientag u. Prüfungsvorbereitung: 50 Stunden		
Qualifikationsziele	Die Studierenden können sicher mit den wichtigsten Komplexitätsklassen umgehen, sie kennen zudem die zu Grunde liegenden Berechnungsmodelle. Die Studierenden haben ein Verständnis für die Grenzen der effizienten Berechenbarkeit erworben und sind in der Lage, Probleme hinsichtlich ihrer algorithmischen Komplexität einzuschätzen und in Komplexitätsklassen richtig einzuordnen.		
Inhalte	<p>In der Komplexitätstheorie beschäftigt man sich damit, welche Probleme mit eingeschränkten Ressourcen (z.B. Zeit oder Speicherplatz) berechnet werden können. Man fasst Probleme dabei zu Komplexitätsklassen zusammen und untersucht deren Beziehung untereinander.</p> <p>Im Kurs werden die Grundlagen der Komplexitätstheorie aus einer algorithmischen Perspektive vermittelt. Als Basistext wird das Buch von Ingo Wegener "Komplexitätstheorie: Grenzen der Effizienz von Algorithmen" verwendet. Der Leittext wird ergänzt mit Übungsaufgaben und Anmerkungen.</p> <p>U.a. werden folgende Themen behandelt:</p> <ul style="list-style-type: none"> <li>- grundlegende Komplexitätsklassen</li> <li>- NP-Vollständigkeit</li> <li>- Interaktive Beweissysteme</li> <li>- probabilistische Komplexitätsklassen</li> <li>- Approximation</li> </ul>		
Inhaltliche Voraussetzung	Grundlagen der theoretischen Informatik, wie sie z.B. im Modul 63912 "Grundlagen der Theoretischen Informatik" (01659) des Bachelorstudiengangs Informatik vermittelt werden.		
Lehr- und Betreuungsformen	Kursmaterial internetgestütztes Diskussionsforum Studientag/e Einsendeaufgaben mit Korrektur und/oder Musterlösung Betreuung und Beratung durch Lehrende		
Anmerkung	Der Basistext muss vor Semesterbeginn beschafft werden. Basistext: Ingo Wegener: Komplexitätstheorie: Grenzen der Effizienz von Algorithmen, Springer, 2003. Nicht zusammen mit dem nicht mehr angebotenen Modul "Grundzüge der Komplexitätstheorie" nutzbar.		
Formale Voraussetzung	keine		
Verwendung des Moduls	M.Sc. Informatik M.Sc. Mathematik M.Sc. Praktische Informatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

bestandene benotete mündliche  
Modulprüfung

Voraussetzung

keine

## Lehrende/r

Jonathan Rollin  
André Schulz

Modulbeauftragte/r André Schulz

Dauer des Moduls  
ein SemesterECTS  
10Workload  
300 StundenHäufigkeit  
in jedem Sommersemester

## Lehrveranstaltung(en)

01684 Effiziente Algorithmen

SS

SWS  
4+2

## Detaillierter Zeitaufwand

Bearbeiten der Kurseinheiten: 210 Stunden  
 Bearbeiten der Übungs- und Einsendeaufgaben: 56 Stunden  
 Studientag u. Prüfungsvorbereitung: 34 Stunden

## Qualifikationsziele

Die Studierenden kennen Paradigmen zum Entwurf von effizienten Algorithmen. Mit Hilfe dieser Paradigmen sind sie in der Lage, effiziente Algorithmen für neue Probleme selbstständig zu finden. Des Weiteren sind ihnen wichtige Algorithmen aus den Gebieten Graphenalgorithmen, Algorithmen für Zeichenketten und Quantenalgorithmen bekannt. Ebenfalls ist es den Studierenden möglich, eine asymptotische theoretische Laufzeitabschätzung vorzunehmen. Die Studierenden kennen zudem Strategien zum Umgang mit NP-schweren Problemen.

## Inhalte

Im Kurs werden die Grundlagen für den Entwurf und die Analyse von effizienten Algorithmen in einem theoretischen Berechnungsmodell vermittelt. Wichtige Entwurfparadigmen werden dazu anhand von Beispielen erklärt. Für viele wichtige Probleme werden effiziente Algorithmen vorgestellt und analysiert. Der Fokus liegt hierbei auf Algorithmen für Zeichenketten, Algorithmen zum Finden von kürzesten Wegen und Algorithmen zur Berechnung von maximalen Flüssen. Ergänzt werden diese Themen durch Überlegungen zum Umgang mit NP-schweren Problemen. Im Kurs wird das theoretische Modell für Quantenalgorithmen vorgestellt. Es werden Phänomene wie Quantenteleportation und einfache Quantenalgorithmen erklärt.

Inhaltliche  
Voraussetzung

--

Lehr- und  
Betreuungsformen

Kursmaterial  
 Einsendeaufgaben mit Korrektur und/oder Musterlösung  
 internetgestütztes Diskussionsforum  
 Studientag/e

## Anmerkung

Keine

## Formale Voraussetzung

keine

## Verwendung des Moduls

M.Sc. Informatik  
 M.Sc. Praktische Informatik

**Prüfungsformen**

Art der Prüfungsleistung

Voraussetzung

## Prüfung

bestandene benotete mündliche  
 Modulprüfung

keine

Stellenwert  
der Note

1/12



# Katalog M2: Computersysteme

Lehrende/r	Jörg M. Haake	Modulbeauftragte/r	Jörg M. Haake
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01802 Betriebssysteme		SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Bearbeitung der Einsendeaufgaben inkl. Verarbeitung des Korrektur-Feedbacks (7 mal 10 Stunden): 70 Stunden Bearbeitung der praktischen Übungen: 10 Stunden Mitwirkung an den Diskussionen in der Kurs-Newsgroup: 20 Stunden Wiederholung und Prüfungsvorbereitung: 60 Stunden		
Qualifikationsziele	Die Teilnehmerinnen und Teilnehmer entwickeln ein vertieftes Verständnis über Aufgaben, Aufbau und die Funktionsweise eines Betriebssystems. Sie können Designentscheidungen für die Anpassung eines Betriebssystems an Anforderungsprofile begründet treffen. Durch die Teilnahme an der Kurs-Newsgroup wird das Einüben wissenschaftlicher Kommunikation gefördert.		
Inhalte	Das Modul behandelt die Funktionsweise von modernen Betriebssystemen aus einer benutzungsorientierten Sicht. Schwerpunktmäßig werden die Dienste vorgestellt, die ein Betriebssystem einem Anwendungsprogramm oder einem Benutzer anbietet. Um diese Dienste verstehen und einschätzen zu können, werden die wichtigsten Prinzipien und Techniken ihrer Realisierung vorgestellt. Themenschwerpunkte sind: Aufgaben und Architektur von Betriebssystemen, Geräteverwaltung und Dateisysteme, Arbeitsspeicherverwaltung, Prozessverwaltung und parallele Prozesse sowie Prozesskommunikation, Sicherheit und Kommandointerpreter. Die Arbeitsweise mit einem Betriebssystem wird in praktischen Übungen vermittelt.		
	<b>Ergänzende Literatur:</b> A. S. Tanenbaum. H. Bos. Modern Operating Systems, Fourth Edition. Prentice Hall, 2014. W. Stallings. Operating Systems: Internals and Design Principles - Ninth Edition. Pearson Education, 2017.		
Inhaltliche Voraussetzung	Teilnehmerinnen und Teilnehmer sollen bereits mit der algorithmischen Denkweise vertraut sein und wenigstens eine höhere Programmiersprache (wie Java, Ada, Modula-2, oder Pascal) kennen, dabei wenigstens ein Betriebssystem (z.B. MS-DOS oder eine UNIX-Variante) zumindest oberflächlich kennengelernt haben, gängige Datenstrukturen wie Listen oder Bäume (z.B. aus dem Kurs Datenstrukturen) kennen und möglichst (aber nicht notwendigerweise) schon Programmieren in Assembler kennengelernt oder auf andere Weise Kenntnisse über Architektur und Funktionsweise digitaler Rechner sowie über den Instruktionssatz einer CPU haben.		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum fachmentorielle Betreuung (Regional- und Studienzentren) Betreuung und Beratung durch Lehrende Studententag/e Zusatzmaterial		
Anmerkung	-		
Formale Voraussetzung	keine		

Verwendung des Moduls M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

**Prüfungsformen**

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete mündliche  
Modulprüfung

keine

Stellenwert 1/12  
der Note

Lehrende/r	Jörg Keller	Modulbeauftragte/r	Jörg Keller
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit regelmäßig
Lehrveranstaltung(en)	01730	Introduction to Information Hiding	SS SWS 2+1
	01731	Network Steganography	WS SWS 2+1
Detaillierter Zeitaufwand	Kurseinheiten: 150 Stunden Einsendearbeiten: 75 Stunden Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	Nach erfolgreicher Teilnahme kennen die Teilnehmerinnen und Teilnehmer die unterschiedlichen Bereiche des Information Hiding und verstehen einfache Verfahren und Gegenmaßnahmen in allen Bereichen. Im Bereich der Netzwerk-Steganografie sind Teilnehmerinnen und Teilnehmer in der Lage, komplexere Verfahren zu verstehen und einfache Verfahren zu implementieren.		
Inhalte	<p>Introduction to Information Hiding (01730): Information Hiding im Kontext der Informatik beschreibt alle Techniken zum Verstecken von digitalen Nachrichten in anderen digitalen Informationen und umfasst im Wesentlichen die Bereiche Kryptografie, Steganografie, Watermarking und Obfuscation. Dabei werden sowohl Methoden zum Verstecken als auch Analyse-Methoden sowie Gegenmaßnahmen betrachtet. Dieser Kurs gibt eine Einführung in die verschiedenen Bereiche des Information Hiding mit Ausnahme der Kryptografie, die in einem eigenen Modul behandelt wird. Aus dem wiederum weiten Feld der Steganografie werden Steganografie in digitalen Medien nebst Steganalysis und Dateisystem-Steganografie behandelt. Die Netzwerk-Steganografie wird in einem eigenen Kurs vertieft behandelt.</p> <p>Network Steganography (01731): Der Kurs zur Netzwerk-Steganografie führt zunächst in die Terminologie und Taxonomie des Themenfeldes ein. Insbesondere werden Konzepte wie direkte/indirekte, rauschende/rauschfreie Netzwerkkanäle, Hiding Patterns und deren Relationen erläutert. Anschließend betrachtet der Kurs Deep Hiding Techniques (DHT), adaptive verdeckte Kanäle, Dead Drops und diverse grundlegende Maßnahmen gegen verdeckte Kanäle in Netzen. Schließlich werden IoT-/CPS-Steganografie und fortgeschrittene Gegenmaßnahmen betrachtet.</p>		
Inhaltliche Voraussetzung	Modul 63512 "Sicherheit im Internet" (01866, 01868) oder Modul 63017 "Datenbanken und Sicherheit im Internet" (01866) sowie deren Voraussetzungen, oder äquivalente Kenntnisse		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung Betreuung und Beratung durch Lehrende internetgestütztes Diskussionsforum Zusatzmaterial Lehrvideos		
Anmerkung	Der Kurs 01730 Introduction to Information Hiding wird erstmalig im SS 2022 angeboten. Die erstmalige Prüfungsteilnahme ist deswegen ab dem SS 2022 möglich.		

Das Modul ist in englischer Sprache.

Formale Voraussetzung keine

Verwendung des Moduls M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/12	Modulprüfung	

Lehrende/r	Lena Oden	Modulbeauftragte/r	Lena Oden
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01728 Virtuelle Maschinen		WS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 150 Stunden Bearbeiten der Einsendearbeiten: 75 Stunden Studientage und Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	Nachdem die Studierenden das Modul bearbeitet haben, können sie die Funktionsweise virtueller Maschinen erklären, die Beschränkungen der Virtualisierung identifizieren, unterschiedliche Arten der Emulation von Befehlssatzarchitekturen beurteilen, die verschiedenen Varianten virtueller Maschinen klassifizieren und gegenüberstellen, selbst virtuelle Maschinen konfigurieren, geeignete Anwendungen kategorisieren und auf einer passenden virtuellen Maschine integrieren, Schwachstellen der Virtualisierung identifizieren, Optimierungsmöglichkeiten entdecken, die Virtualisierung in geeigneter Form reorganisieren und schließlich für die jeweilige Anwendung eine bestmöglich angepasste virtuelle Plattform produzieren.		
Inhalte	Eine virtuelle Maschine emuliert ein komplettes Computersystem durch Softwarekomponenten, die einen einheitlichen Zugang zur Hardware eines realen Computersystems bereitstellen. Durch die Virtualisierung wird es möglich, auf ein und demselben Computersystem nacheinander oder auch gleichzeitig mehrere verschiedene Betriebssysteme laufen zu lassen. Die Virtualisierung von Computersystemen bietet eine Fülle von Vorteilen, wie z. B. die gleichzeitige Nutzung von mehreren Betriebssystemen, den einfachen und kostengünstigen Aufbau von Testumgebungen und die verbesserte Auslastung von Mehrkern-Prozessoren. Im Rahmen dieses Kurses werden die Grundlagen heutiger Virtualisierungstechniken herausgearbeitet. Der Kurs basiert auf dem Buch "Virtual Machines" von Smith und Nair, erschienen bei Elsevier 2005. Zu diesem englischsprachigen Basistext erhalten die Studierenden einen deutschsprachigen Leittext.		
	<b>Ergänzende Literatur:</b> I. D. Craig, Virtual Machines, Springer-Verlag, New York, 2005		
Inhaltliche Voraussetzung	Kenntnisse aus den Modulen 63013 "Computersysteme" (01608/01609) sowie 63012 "Softwaresysteme", Kurs "Betriebssysteme und Rechnernetze" (01801)		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Studientag/e		
Anmerkung	Der Basistext muss vor Semesterbeginn beschafft werden. Basistext: J. E. Smith and R. Nair, Virtual Machines, Elsevier, 2005		
Formale Voraussetzung	keine		
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

bestandene benotete mündliche  
Modulprüfung

Voraussetzung

keine

Lehrende/r	Lena Oden	Modulbeauftragte/r	Lena Oden
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01729 Advanced Parallel Computing		SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 150 Stunden, Bearbeiten der Einsendearbeiten: 75 Stunden, Studientage und Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	Nachdem die Studierenden das Modul bearbeitet haben, können sie Modellierungswerkzeuge für parallele Programme erklären, Scheduling-Algorithmen für homogene und heterogene Zielsysteme im Umfeld statischer und dynamischer Anwendungsszenarien klassifizieren, die Auslastung von Mehrkernprozessoren analysieren und optimieren, Programmieretechniken für hybride Zielarchitekturen anwenden, die Beschränkungen der Virtualisierung identifizieren, unterschiedliche Arten der Emulation von Befehlssatzarchitekturen beurteilen, die verschiedenen Varianten virtueller Maschinen klassifizieren und gegenüberstellen, selbst virtuelle Maschinen konfigurieren, geeignete Anwendungen kategorisieren und auf einer passenden virtuellen Maschine integrieren, Schwachstellen der Virtualisierung identifizieren, Optimierungsmöglichkeiten entdecken, die Virtualisierung in geeigneter Form reorganisieren und schließlich für die jeweilige Anwendung eine bestmöglich angepasste virtuelle Plattform produzieren.		
Inhalte	Im Kurs werden zunächst Modellierungswerkzeuge für parallele Programme eingeführt. Darauf aufbauend werden statische und dynamische Scheduling-Verfahren vorgestellt, die bei hochperformanten Parallelrechnern eine automatisierte Zuordnung der Tasks zu den einzelnen Prozessoren ermöglichen. Weiterhin werden Programmieretechniken für innovative parallele Architekturen eingeführt. Hierbei wird auch ausführlich auf die Programmierung von Graphical Processing Units (GPUs) eingegangen. In der Praxis werden parallele Implementierungen vor allem zur Lösung komplexer Optimierungsprobleme benötigt. Daher werden sowohl Approximationsverfahren als auch Heuristiken für numerische und kombinatorische Problemstellungen ausführlicher behandelt und analysiert. Schließlich wird anhand von Beispielen deren Einsatz im Umfeld von Forschung und Industrie vorgestellt.		
	<p><b>Ergänzende Literatur:</b></p> <p>E. Alba, Parallel Metaheuristics, Wiley, 2005</p> <p>C. Bishof et al., Parallel Computing, Architectures, Algorithms, and Applications, IOS Press, 2008</p> <p>C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surv. , 35(3):268–308, September 2003.</p> <p>T. G. D. Kirk and Wen-mei Hwu, Programming Massively Parallel Processors – A Hands-on Approach, Morgan Kaufman, 2010</p> <p>M. Drozdowski, Scheduling for Parallel Processing, Springer-Verlag, 2009</p> <p>Yu-Kwong Kwok and I. Ahmad. Static scheduling algorithms for allocating directed task graphs to multiprocessors. ACM Comput. Surv. , 31(4):406–471, December 1999.</p>		
Inhaltliche Voraussetzung	Kenntnisse der Inhalte des Moduls 63712 "Parallele Programmierung" (01727)		
Lehr- und Betreuungsformen	Kursmaterial internetgestütztes Diskussionsforum Einsendeaufgaben mit Korrektur und/oder Musterlösung		



Betreuung und Beratung durch Lehrende

Anmerkung

-

Formale Voraussetzung

keine

Verwendung des Moduls

M.Sc. Informatik

M.Sc. Praktische Informatik

M.Sc. Wirtschaftsinformatik

### Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete mündliche

keine

Stellenwert  
der Note

1/12

Modulprüfung

63715

## PC-Technologie

Lehrende/r	Wolfram Schiffmann	Modulbeauftragte/r	Wolfram Schiffmann
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01744 PC-Technologie		SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Einüben des Stoffes (insbesondere durch Einsendeaufgaben (7 mal 15 Stunden): 105 Stunden Wiederholung und Prüfungsvorbereitung (Selbststudium): 55 Stunden		
Qualifikationsziele	Die Studierenden erwerben die Fähigkeiten, das komplexe Zusammenwirken der Komponenten in einem Personal Computer (PC) zu verstehen. Insbesondere wird der Aufbau der x86-Prozessoren im Detail erlernt. Daneben werden tief gehende Kenntnisse der Speicherorganisation und -verwaltung erworben. Sie lernen Begriffe und Abkürzungen kennen, die im Alltags- und Berufsleben häufig benutzt werden, ohne dass ihre Bedeutung hinterfragt wird. Die Beschreibung der wichtigsten Peripheriegeräte (Monitore, Tastatur, Audiosysteme, Drucker und Projektoren) zeigt ihnen, wie die täglich benutzten Geräte funktionieren und wo deren Stärken und Schwächen liegen. Weiterhin verstehen sie nach der Bearbeitung des Kurses die heute übliche Einbindung eines Personal Computers in ein Netzwerk, insbesondere das Internet, sowie die dafür benötigten Koppeleinheiten und Protokolle. Ein eigenes Kapitel macht sie mit den Besonderheiten der sich mobilen Varianten eines PCs vertraut.		
Inhalte	Im Kurs werden der Aufbau und die Funktionsweise eines PCs vorgestellt. Neben Prozessoren von AMD und Intel werden Mainboards, Chipsätze, Ein-/Ausgabe-Busse, Speicher- und Prozessverwaltung, Speichermedien, Audio- und Graphikschnittstellen sowie die wichtigsten Peripheriegeräte im Detail vorgestellt. Einen weiteren Schwerpunkt bildet die Behandlung von Netzwerkschnittstellen. Neben Kupfer- und Lichtwellenleitern kommen dabei verstärkt Techniken zur drahtlosen Kommunikation zum Einsatz. Damit können auch mobile Geräte leicht in die Netzwerke integriert werden. Wegen ihrer wachsenden Bedeutung wird ebenfalls in die Technik mobiler Computer eingeführt. Der Kurs vermittelt neben der Hardware-Technologie auch Einblicke in die benötigte System-Software. So wird beispielsweise in die Grundlagen von Dateisystemen und Gerätetreibern eingeführt.		
	<b>Ergänzende Literatur:</b> H. Messmer, K. Dembrowski: PC Hardwarebuch. Aufbau, Funktionsweise, Programmierung, Addison-Wesley, 2003 J. Ortmann: Einführung in die PC-Grundlagen. Addison-Wesley, 2006, ISBN: 978-3-8273-2339-2 S. Mueller: PC-Hardware Superbibel. Markt und Technik-Verlag, 2005, ISBN: 978-3827268174		
Inhaltliche Voraussetzung	Grundkenntnisse in Mikrocomputer-Technik, insbesondere in Architektur und Programmierung von Mikroprozessoren, wie sie z.B. in Modul 63711 "Anwendungsorientierte Mikroprozessoren" (01706) gelehrt werden.		
Lehr- und Betreuungsformen	Kursmaterial internetgestütztes Diskussionsforum		

Einsendeaufgaben mit Korrektur und/oder Musterlösung  
Betreuung und Beratung durch Lehrende

Anmerkung

-

Formale Voraussetzung

keine

Verwendung des Moduls

M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

### Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete mündliche  
Modulprüfung

keine

Stellenwert  
der Note

1/12

Lehrende/r	Herwig Unger	Modulbeauftragte/r	Herwig Unger
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01690	Kommunikations- und Rechnernetze	WS/SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Einüben des Stoffes (insbesondere durch Einsendeaufgaben (7 mal 15 Stunden): 105 Stunden Wiederholung und Prüfungsvorbereitung (Onlineveranstaltungen und Selbststudium): 55 Stunden		
Qualifikationsziele	Die Studierenden entwickeln Vertrautheit mit grundlegenden Fragen zu Struktur und Eigenschaften unterschiedlicher Netzwerksysteme. Sie erwerben die Fähigkeit, solche Systeme nach verschiedenen Gesichtspunkten zu klassifizieren und deren Einsatzgebiete und Grenzen zu benennen. Sie erwerben weiterhin die Fähigkeit, Signale anhand grundlegender Eigenschaften zu analysieren und zu klassifizieren. Weiterhin werden sie in die Lage versetzt, bestehende Systeme in der Praxis auf wesentliche Eigenschaften wie bspw. Sicherheit und Robustheit hin zu untersuchen.		
Inhalte	Das Modul bietet einen umfassenden Überblick über die Aufgaben, Problemstellungen und Lösungen, die bei der Netzwerkkommunikation auftreten. Es werden die signaltheoretischen Grundlagen vorgestellt und die wesentlichen Protokolle und Anwendungen in heutigen Weitverkehrsnetzen detailliert beschrieben. Weiterhin bietet das Modul eine fundierte Einführung in Modelle zur Beschreibung und Analyse verschiedenster Netzwerkstrukturen. Dazu gehören unter anderem Small Worlds, Zufallsnetze und skalenfreie Netze. Schließlich werden beispielhaft einige in der Praxis auftretende Netzwerksysteme vorgestellt. Dazu gehören Bezahlsysteme, RFID und Peer-to-Peer-Systeme.		
	<b>Ergänzende Literatur:</b> A. S. Tanenbaum: Computernetzwerke, 4. überarbeitete Auflage, Pearson 2003 P. Mahlmann, Chr. Schindelbauer: P2P Netzwerke: Algorithmen und Methoden, Springer 2007 Frey, Bossert, Fliege: Signal- und Systemtheorie, 2. korrigierte Auflage, Vieweg +Teubner 2009		
Inhaltliche Voraussetzung	Grundkenntnisse; z.B. aus Modul 63012 "Softwaresysteme" (01801)		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung Betreuung und Beratung durch Lehrende internetgestütztes Diskussionsforum Studientag/e Lehrvideos		
Anmerkung	-		
Formale Voraussetzung	keine		
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik		

M.Sc. Wirtschaftsinformatik

**Prüfungsformen**

Prüfung

Stellenwert der Note 1/12

Art der Prüfungsleistung

bestandene benotete Prüfungsklausur

Voraussetzung

keine

Lehrende/r	Marcel Schaible	Modulbeauftragte/r	Herwig Unger	
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden	Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01711	Fehlertoleranz in Computersystemen und Netzwerken		WS/SS SWS 2+1
	01867	Sicherheit im Internet II		WS/SS SWS 2+1
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 140 Stunden Bearbeiten der Übungsaufgaben: 70 Stunden Onlineveranstaltungen und Prüfungsvorbereitung: 90 Stunden			
Qualifikationsziele	<p>Die Studierenden erwerben ein vertieftes Verständnis konkreter Angriffe und Gefahren im Internet. Sie sind in der Lage, Schutzmaßnahmen für einzelne Rechner sowie für Netzwerke zu beurteilen, auch im rechtlichen Kontext, sowie bei der Implementierung solcher Schutzmaßnahmen mitzuwirken.</p> <p>Die Studierenden erwerben grundlegende theoretische Grundlagen auf dem Gebiet der Fehlertoleranz. Sie haben gelernt, dieses Kenntnisse in komplexen Methoden auf praktische Fragestellungen anzuwenden, wobei insbesondere Beispiele aus dem Bereich der Rechnerarchitektur und Kommunikationsnetze im Zielpunkt der Wissensvermittlung stehen.</p>			
Inhalte	<p>01867 vertieft das Thema Sicherheit im Internet und baut dazu auf dem Kurs 01866 "Sicherheit im Internet I" auf. Er besteht aus vier Teilen. Im ersten Teil werden typische Angriffe auf Systeme genauer vorgestellt. Hierzu gehören die Angriffe auf Schwächen in Protokollen ebenso wie Angriffe auf die Konfiguration von Systemen. Spezielle Überwachungs-, bzw. Angriffsprogramme werden vorgestellt. Den Abschluss des ersten Teils bildet ein Abschnitt zu Angriffen auf Verschlüsselungsalgorithmen. Der zweite Teil beschäftigt sich mit der Benutzersicherheit. Ausgehend von typischen E-Commerce Transaktionen werden verschiedene Möglichkeiten zur Bezahlung im Internet mit ihren Eigenschaften vorgestellt. Insbesondere anonyme Bezahlverfahren, also elektronisches Bargeld wird mit den dahinter stehenden Prinzipien diskutiert. Außerdem werden in dieser Kurseinheit einige eher mathematische Aspekte des Thema Sicherheit betrachtet: Advanced Hashing, Primzahltest und Zufallszahlen. Der dritte Teil beschäftigt sich mit der Anbietersicherheit. Konkret geht es um den Aufbau eines privaten Netzes über öffentliche Netzwerke. Dies wird durch Virtual Private Networks (VPN) realisiert. Ihre Basistechnologien und ihr Aufbau werden vorgestellt. Außerdem werden Intrusion Detection Systeme (IDS) zur Unterstützung bei der Erkennung von Angriffen besprochen. Es werden wieder die zugrunde liegenden Techniken und die Eigenschaften von IDS besprochen. Im letzten Teil des Kurses geht es um die Erstellung von sicheren Systemen. Zunächst werden die Rahmenbedingungen (wie beispielsweise gesetzliche Vorschriften) vorgestellt, die Einfluss auf die Eigenschaften sicherer Systeme haben. Anschließend werden Hinweise zu Software Engineering Prozessen gegeben. Die Beachtung dieser Hinweise vereinfacht die Erstellung sicherer Systeme.</p> <p>01711 untersucht nach einer Klärung wichtiger Begriffe sowie der Unterschiede zwischen Fehlertoleranz und Risikominimierung die grundlegenden Arten von Fehlern und Prinzipien der Fehlertoleranz. Dies umfasst zunächst die Behandlung der theoretischen Grundlagen, die im Wesentlichen ein Basiswissen der Wahrscheinlichkeitsrechnung und der Statistik umfassen. Darauf aufbauend werden Ansätze vorgestellt, die zu einer fehlertolerierenden Datenspeicherung und Datenübertragung beitragen. Ein letzter Kursschwerpunkt fokussiert auf die Entwicklung fehlertoleranter Kommunikationssysteme und verteilter Rechnersysteme mittels protokollbasierter Ansätze.</p>			

### Ergänzende Literatur:

C. Eckert: IT-Sicherheit. 2. Auflage. Oldenbourg 2003

R. Oppliger: Security Technologies for the World Wide Web 2nd Edition, Artech House 2003

B. Schneier: Applied Cryptography 2nd Edition, Wiley 1996

Bacon, J. and Harris, T. (2003). Operating Systems Concurrent and Distributed Software Design. NY: Addison-Wesley.

Bertsekas, D. P. and Gallager, R. (1987). Data Networks (2nd Ed.). NJ: Pearson Prentice Hall.

Devore, J. L. (2007). Probability and Statistics for Engineering and the Sciences. CA: Brooks/Cole Publishing Company

Koren, I. and Mani Krishna, C. (2007). Fault-Tolerant Systems. CA: Elsevier.

Tanenbaum, A. S. (2008). Modern Operating Systems (3rd Ed.). NJ: Pearson Prentice Hall.

### Inhaltliche Voraussetzung

Kurs 01867: Modul 63512 "Sicherheit im Internet" (01866) oder Modul 63017 "Datenbanken und Sicherheit im Internet" (01866) sowie dessen Voraussetzungen, oder äquivalente Kenntnisse

### Lehr- und Betreuungsformen

Kurs 01711: Grundkenntnisse; z.B. aus Modul 63012 "Softwaresysteme" (01801)

Kursmaterial

Einsendeaufgaben mit Korrektur und/oder Musterlösung

Betreuung und Beratung durch Lehrende

internetgestütztes Diskussionsforum

### Anmerkung

Früher Kursnummer 21811, neue Kursnummer 01711

Das Modul kann letztmalig im SS 2022 belegt werden. Eine Prüfungsteilnahme ist nur noch bis einschließlich SS 2023 möglich.

### Formale Voraussetzung

keine

### Verwendung des Moduls

M.Sc. Informatik

M.Sc. Praktische Informatik

M.Sc. Wirtschaftsinformatik

### Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete Prüfungsklausur

keine

Stellenwert  
der Note

1/12

# Katalog M3: Informationssysteme



Lehrende/r	Jörg M. Haake	Modulbeauftragte/r	Jörg M. Haake
	Dauer des Moduls ein oder drei Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01880	Computerunterstütztes kooperatives Arbeiten (CSCW)	SS SWS 2+1
	01883	Computerunterstütztes kooperatives Lernen (CSCL)	SS SWS 2+1
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (8 mal 20 Stunden): 160 Stunden Bearbeitung der Einsendeaufgaben (8 mal 5 Stunden): 40 Stunden Mitwirkung an den Diskussionen in der Kurs-Newsgroup: 30 Stunden Wiederholung und Prüfungsvorbereitung: 70 Stunden		
Qualifikationsziele	Die Studierenden entwickeln ein vertieftes Verständnis über die interdisziplinären Grundlagen und Forschungsmethoden im Forschungsgebiet CSCW/CSCL. Sie können begründete Entwurfsentscheidungen bei der Entwicklung von CSCW/CSCL-Systemen treffen und bestehende CSCW/CSCL-Systeme bzgl. Ihrer Verwendbarkeit in verschiedenen Anforderungssituationen auswählen und bewerten. Sie können sich selbstständig interdisziplinäre Arbeiten erschließen und ein übergreifendes Begriffsverständnis erarbeiten. Darüber hinaus können sie die Ergebnisse von Evaluationsstudien bezüglich ihrer Aussagekraft beurteilen. Durch die Teilnahme an Internet-Diskussionsgruppen wird das Einüben wissenschaftlicher Kommunikation gefördert.		
Inhalte	Der Kurs 01880 behandelt verschiedene Aspekte der Rechnerunterstützung für Gruppen von Personen, die gemeinsam an einem Problem arbeiten. Nach den Grundlagen des kooperativen Arbeitens werden die im 3K-Modell definierten Systemklassen, Kommunikation, gemeinsame Informationsräume, Workflow Management und Workgroup Computing betrachtet und an Beispielen illustriert. Abschließend widmet sich der Kurs der Realisierung von synchroner Groupware. Der Kurs 01883 behandelt das Thema des computerunterstützten kooperativen Lernens (CSCL) aus interdisziplinärer Sicht. Er bietet eine anwendungsorientierte Einführung in das Forschungsgebiet CSCL und soll Erfahrung in der selbstständigen Einarbeitung in ein interdisziplinäres Forschungsgebiet vermitteln. Dazu werden die Themenbereiche Grundlagen, Werkzeuge, Didaktische Konzepte sowie Umsetzungen und Perspektiven von CSCL behandelt.		
	<b>Ergänzende Literatur:</b> S. Teufel, C. Sauter, T. Mühlherr und K. Bauknecht: Computerunterstützung für die Gruppenarbeit . Addison-Wesley, 1995. U.M. Borghoff und J. H. Schlichter: Computer-Supported Cooperative Work. Springer-Verlag Berlin Heidelberg New York, 2000. G. Schwabe, N. Streitz und R. Unland (Hrsg.): CSCW-Kompodium. Springer-Verlag Berlin Heidelberg New York, 2001. T. Schümmer und St. Lukosch: Patterns for Computer-Mediated Interaction. John Wiley & Sons, Ltd., 2007.		
Inhaltliche Voraussetzung	Grundkenntnisse in Programmierung. Grundkenntnisse Betriebssysteme (Prozesse) und Rechnernetze (Protokolle), z.B. aus Modul 63012 "Softwaresysteme (01801)". Kenntnisse in Modul 63211 "Verteilte Systeme" (01678) sind hilfreich.		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende		

Anmerkung Der Basistext muss vor Semesterbeginn beschafft werden. Basistext zu 01883: Haake, G. Schwabe, M. Wessner (Hrsg.): CSCL-Kompendium 2.0 - Lehr- und Handbuch zum computerunterstützten kooperativen Lernen. Oldenbourg Verlag, München, 2012.

Formale Voraussetzung keine

Verwendung des Moduls M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/12	Modulprüfung	

Lehrende/r	Niels Seidel	Modulbeauftragte/r	Jörg M. Haake
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01884 Gestaltung Kooperativer Systeme		WS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Bearbeitung der Übungen in der Kurs-Lernumgebung (Moodle): 50 Stunden Diskussion in virtuellen Präsenzveranstaltungen und in den Foren der Kurs-Lernumgebung (Moodle): 40 Stunden Auseinandersetzung mit Lösungen von Kurs-Teilnehmenden (Peer-Review): 10 Stunden Wiederholung für Prüfungsvorbereitung respektive Erstellung der Hausarbeit: 60 Stunden		
Qualifikationsziele	Die Teilnehmenden entwickeln ein vertieftes Verständnis für das Design von kooperativen Systemen, sowohl Kenntnisse auf technischer Gestaltungsebene als auch auf sozio-technischer Ebene. Studierende können nach Abschluss des Moduls Kooperationsprozesse analysieren und auf kooperative Systeme abbilden. Sie werden in die Lage versetzt, Prototypen visuell zu beschreiben und die Interaktionsprozesse zu formulieren. Neben diesen fachspezifischen Zielen verbessern die Studierenden wichtige Schlüsselqualifikationen: Verteilte Kooperation über das Internet, Verständnis von Gruppenprozessen und Gruppeninteraktion, Erschließung englischer Fachliteratur und Bewertung der Folgen des Einsatzes von sozio-technischen Systemen.		
Inhalte	In diesem Kurs werden Gestaltungskonzepte kooperativer Systeme anhand von Entwurfsmustern vermittelt. Diese werden in den Übungen am Entwurf eines konkreten kooperativen Systems in Form von visuellen Skizzen der Benutzungsschnittstellen und textuellen Beschreibungen der Anwendungsfälle praktisch eingeübt. Neben dem gestalterischen Aspekt betrachten die Teilnehmenden des Moduls philosophische und ethische Grundlagen kooperativer Systeme. So wird z. B. diskutiert, wie Identität in virtuellen Gemeinschaften herausgebildet, wie Gruppenprozesse unterstützt, Kommunikation gestaltet und wie gegenseitige Wahrnehmung in kooperativen Systemen hergestellt und in den Arbeitsalltag der Nutzenden eingebettet werden kann. Hierauf aufbauend werden die Teilnehmenden in die Lage versetzt, ein kooperatives System sowohl aus Benutzungssicht als auch aus technischer Sicht zu gestalten. Dabei werden Entwurfsmuster als zentrales Werkzeug eingesetzt. Diese sind auf verschiedenen Abstraktionsebenen angesiedelt (Virtuelle Gemeinschaft, Arbeitsgruppe, Gruppenbewusstsein und Technologie) und erlauben einen ganzheitlichen Blick auf computerunterstützte virtuelle Gemeinschaften und Kleingruppen. Der Schwerpunkt liegt dabei auf der mobilen computervermittelten Interaktion mit Hilfe kooperativer Systeme.		
	<b>Ergänzende Literatur:</b> T. Schümmer und S. Lukosch: Patterns for Computer-Mediated Interaction. John Wiley & Sons, Ltd., 2007 S. Greenberg, S. Carpendale, N. Marquardt und B. Buxton: Sketching User Experiences: The Workbook. Elsevier, 2011. C. Crumlish und E. Malone: Designing Social Interfaces. O'Reilly, 2009. T. Gross und M. Koch: Computer-Supported Cooperative Work, Odenbourg, 2007. T. Neil: Mobile Design Pattern Gallery. O'Reilly, 2014. J. Donath: The social machine: designs for living online. MIT Press, Cambridge, Mass (2014).		
Inhaltliche Voraussetzung	Englische Sprachkenntnisse, grundständige Programmierkenntnisse.		

Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung Betreuung und Beratung durch Lehrende Video-Meetings Zusatzmaterial internetgestütztes Diskussionsforum Studientag/e	
Anmerkung	Das Modul wird im Wintersemester in Form einer Hausarbeit und im Sommersemester in Form einer mündlichen Prüfung geprüft.	
Formale Voraussetzung	keine	
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik	
<b>Prüfungsformen</b>	<b>Art der Prüfungsleistung</b>	<b>Voraussetzung</b>
Prüfung	s. Anmerkung	Für die Teilnahme an der mündlichen Prüfung setzen wir die korrekte Bearbeitung von 48 % der Einsendeaufgaben voraus.
Stellenwert der Note	1/12	

Lehrende/r	Matthias Hemmje	Modulbeauftragte/r	Matthias Hemmje
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01870	Informationsvisualisierung im Internet	SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeitung der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Einüben der Inhalte (7 mal 15 Stunden): 105 Stunden Wiederholung und Prüfungsvorbereitung: 55 Stunden		
Qualifikationsziele	Die Studierenden beherrschen die wichtigsten allgemeinen Begriffe, Modelle und Methoden der Informationsvisualisierung und können diese innerhalb exemplarischer Technologien und Anwendungen der Informationsvisualisierung zuordnen sowie innerhalb eigener Entwürfe und Modellierungen von Informationsvisualisierungen in Benutzungsschnittstellen zu Informationssystemen anwenden.		
Inhalte	Die adäquate Repräsentation und visuelle Präsentation von abstrakten Daten bildet eine wichtige Grundlage für deren effektive Erfassung, deren Verständnis und deren Nutzung, z. B. zum Zwecke der Information oder der Entscheidungsunterstützung. Die Verwendung geeigneter Modelle, Methoden, Visualisierungstechniken und korrespondierender technischer Standards zur Unterstützung automatisierter Informationsvisualisierung erleichtert dabei die Entwicklung von Informationsvisualisierungssoftware, von visuellen Benutzungsschnittstellen zu Informationssystemen und ganz generell von maschinenlesbaren und maschinell darstellbaren und nutzbaren Informationsvisualisierungen sowie deren Wahrnehmung, Bearbeitung, Nutzung und Austausch zwischen Menschen, zwischen Menschen und Maschinen sowie zwischen Maschinen. Solche Grundlagen stellen ein gemeinsames Grundverständnis unabhängig von einer individuellen Ausgestaltung einer Informationsvisualisierungsanwendung sicher und ermöglichen damit auch die Realisierung interaktiver, dynamischer und damit zum einen schnell wechselnder bzw. sich schnell verändernder Anwendungen von Informationsvisualisierungen, die in unserer durch das Internet global vernetzten und sich kontinuierlich veränderten Welt immer weiter an Bedeutung gewinnen.		
	Ergänzende Literatur: Colin Ware: Information Visualization. Morgan Kaufmann, 2004 R.Carey, G. Bell: The Annotated Vrm1 2.0 Reference Manual, Addison Wesley, 1997 D. Brutzman, L. Daly: X3D. Extensible 3D Graphics for Web Authors, Morgan Kaufmann, 2007		
Inhaltliche Voraussetzung	keine		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung Betreuung und Beratung durch Lehrende		
Anmerkung	-		
Formale Voraussetzung	keine		
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik M.Sc. Wirtschaftsinformatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

bestandene benotete Prüfungsklausur

Voraussetzung

Die Zulassung zur Klausur kann nur erfolgen, wenn bei 5 von 7 Kurseinheiten mindestens 50 % der möglichen Punkte in den Einsendeaufgaben erreicht wurden.

Lehrende/r	Matthias Hemmje	Modulbeauftragte/r	Matthias Hemmje
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01877    Dokumenten- und Wissensmanagement im Internet		WS/SS    SWS 4+2
Detaillierter Zeitaufwand	Bearbeitung der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Einüben der Inhalte (7 mal 15 Stunden): 105 Stunden Wiederholung und Prüfungsvorbereitung: 55 Stunden		
Qualifikationsziele	<p>Das hauptsächliche Ziel dieses Kurses ist der Erwerb von praktischer Kompetenz und Handlungsfähigkeit.</p> <p>Die Studierenden sind nach Abschluss des Kurses in der Lage, das Modell der strukturierten Dokumente zu beschreiben und zu beurteilen. Sie können die wichtigsten Begriffe, Modelle und Methoden der Anwendung im Kontext von wissensbasierten Informationssystemanwendungen für das Internet formulieren.</p> <p>Teilnehmende des Kurses erhalten Kenntnisse über HTML, CSS und JavaScript und verstehen es, entsprechenden Code zu analysieren und eigenständig zu erstellen.</p> <p>Weiterhin erschließen sich die Studierenden die grundlegenden Strukturen von XML-Dokumenten, sowie den zugehörigen Schemavarianten DTD und XSD, und können auf dieser Basis eigene Codebeispiele generieren.</p> <p>Die Studierenden erarbeiten die grundlegende Funktionsweise und den Aufbau von XSLT Templates, sowie Suchanfragen an XML-Informationquellen mittels XQuery. Sie sind in der Lage, Anfragen an XML-Dokumente mit XQuery selbständig zu formulieren.</p> <p>Als Basis des Themengebietes Semantisches Web beschäftigen sich die Studierenden im Folgenden mit dem Begriff der Semantik und dem sogenannten Resource Description Framework (RDF).</p> <p>Abschließend wird den Studierenden ein Einblick in Aufbau und Anwendungsgebiete von Ontologiesprachen für das Internet vermittelt. Sie entwickeln ein Verständnis für die Rolle von Ontologien bei der Informationsintegration.</p>		
Inhalte	Kurseinheit 1 – Das Modell der strukturierten Dokumente Kurseinheit 2 – Einführung in HTML, CSS und JavaScript Kurseinheit 3 – Einführung in XML, DTD und XSD Kurseinheit 4 – Einführung in XSLT und XQuery Kurseinheit 5 – Semantik und Semantisches Web Kurseinheit 6 – Semantische Integration Kurseinheit 7 – Ontologiesprachen für das Internet		
	<p>In diesem Kurs werden allgemeine Formate für Dokumente sowie für strukturierte Dokumente, die im Internet verfügbar sind, vorgestellt. Es folgt eine detaillierte Betrachtung des Modells der strukturierten Dokumente, in der die Studierenden verschiedene Arten von Textdokumenten sowie Umsetzungsmöglichkeiten des Modells analysieren.</p> <p>Weiter untersuchen sie verschiedene Modelle von Zeichenkodierungen, die den Inhalt von Dokumenten in maschinenlesbarer Form repräsentieren, sowie deren einzelnen Bestandteile.</p> <p>Die Teilnehmenden des Kurses befassen sich mit der Historie der Markup-Sprachen und sind in der Lage, deren Entwicklung mit der Hypertext-Markup-Language (HTML) als Ausgangsbasis zu erläutern.</p> <p>Die Anbindung von Stylesheets an HTML-Dokumente, die CSS-Syntax und die Verwendung von CSS-Selektoren werden ebenfalls dargestellt und können anhand von Praxisbeispielen übertragen und vertieft werden.</p>		

Im Anschluss wird im Ansatz auf JavaScript als Webprogrammiersprache eingegangen und die grundlegende Funktionsweise beleuchtet.

Die Studierenden setzen sich des Weiteren detailliert mit XML als einer Meta-Auszeichnungssprache, mit der andere Auszeichnungssprachen definiert werden können, sowie als Datenformat und insbesondere auch als Datenaustauschformat zwischen Anwendungen auseinander.

Die Studierenden lernen Dokumenttyp-Definitionen (kurz DTDs) und XML-Schema-Definitionen (kurz XSDs) kennen. Die Teilnehmenden des Kurses sind in der Lage, eigene Schemabeschreibungen für XML-Dokumente zu erstellen.

Anknüpfend an die Grundlagen von XML werden die Anwendungen XSLT und XQuery erarbeitet. Die Studierenden des Kurses können eigene XSLT-Templates, beispielsweise mit dem Ziel der selektiven Datenausgabe, erzeugen. Darauffolgend wird XQuery als Abfragesprache für XML-Dokumente eingeführt, mittels derer Suchanfragen auf XML-Datenquellen gestellt werden können.

Die Studierenden setzen sich weiterhin mit dem semantischen Web als Fortentwicklung des bestehenden WWWs auseinander. Betrachtet werden die Bereitstellung von Informationsquellen im Hinblick auf menschliche und auch maschinenlesbare Nutzung, sowie Probleme und Lösungsansätze bei der Etablierung eines Semantischen Netzes.

Die Studierenden analysieren das Resource Description Framework (RDF) als ein Modell der Wissensrepräsentation und der Darstellung von Metadaten.

Als Ausgangspunkt des Ontologie-basierenden Informationsmanagements differenzieren die Studierenden zwischen drei Konzepten der semantischen Heterogenität. Der Begriff der Ontologie, sowie ihr Aufbau, auch im Vergleich zur Taxonomie wird näher betrachtet und ihre Rolle bei der Reduktion von semantischer Heterogenität durch Konzeptualisierungen und gemeinsam genutztes Vokabular beleuchtet.

Des Weiteren werden der Übersetzungsprozess und das Ontologie-Alignment betrachtet, wodurch die Interoperabilität zwischen semantischen (Web-)Applikationen gewährleistet werden soll. Die Begriffe Ontology Mapping und Ontology Merging sind nach der Bearbeitung ebenfalls bekannt und können kontrastiert werden.

#### Ergänzende Literatur:

P. Aiken, M. D. Allen, "XML in Data Management: Understanding and Applying Them Together", Morgan Kaufmann, June 7, 2004, ISBN-13: 978-0120455997

S. Abiteboul, P. Bunemann, D. Suciu: "Data on the Web: From Relations to Semistructured Data and XML" Morgan Kaufmann, October 12, 1999, ISBN-13: 978-1558606227

G. Antoniou, F. van Harmelen: "A Semantic Web Primer", The MIT Press, April 1, 2004, ISBN-13: 978-0262012102

J. Davies, R. Studer, P. Warren (eds): "Semantic Web Technologies: Trends and Research in Ontology-based Systems", Wiley, July 11, 2006, ISBN-13: 978-0470025963

A. Sheth, M. Lytras: "Semantic Web-Based Information Systems: State-of-the-Art, Applications" Cybertech Publishing, December 11, 2006, ISBN-13: 978-1599044279

Inhaltliche  
Voraussetzung

Programmierkenntnisse

Lehr- und  
Betreuungsformen

Kursmaterial

Einsendeaufgaben mit Korrektur und/oder Musterlösung

internetgestütztes Diskussionsforum

Betreuung und Beratung durch Lehrende



Anmerkung -  
Formale Voraussetzung keine

Verwendung des Moduls M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

### Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete Prüfungsklausur

Die Zulassung zur Klausur kann nur erfolgen, wenn bei 5 von 7 Kurseinheiten mindestens 50 % der möglichen Punkte in den Einsendeaufgaben erreicht wurden.

Stellenwert 1/12  
der Note

Lehrende/r	Matthias Hemmje	Modulbeauftragte/r	Matthias Hemmje
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit s. Anmerkung
Lehrveranstaltung(en)	01875    Multimedialinformationssysteme I		SWS 2+1
	01876    Multimedialinformationssysteme II		SWS 2+1
Detaillierter Zeitaufwand	Bearbeitung der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Einüben der Inhalte (7 mal 15 Stunden): 105 Stunden Wiederholung und Prüfungsvorbereitung: 55 Stunden		
Qualifikationsziele	<p>Die Studierenden beherrschen die wichtigsten allgemeinen Begriffe, Modelle und Methoden der aktuellen Technologien sowie den gegenwärtigen Stand der Forschung und Entwicklung in den relevanten Themengebieten Multimediatechnologien sowie Hypermedia- und Multimedia-Information-Retrieval-Methoden und den dazu korrespondierenden Technologien und können diese zuordnen und innerhalb exemplarischer Technologie- und Anwendungsentwicklungen in den relevanten Themengebieten Multimediatechnologien sowie zu Hypermedia- und Multimedia-Information-Retrieval-Methoden anwenden.</p> <p>Die Studierenden beherrschen die wichtigsten Beispiele, Problemschilderungen, Erfahrungsberichte und Best-Practice-Lösungsansätze zur Sicherung der langfristigen Verfügbarkeit digitaler nicht-textueller Medien.</p>		
Inhalte	<p>Kurs 01875: Die Repräsentation, Speicherung, Verwaltung und Verarbeitung großer Mengen von Multimedia-Dokumenten, die nicht nur aus Texten bestehen, sondern Graphiken, Fotos sowie Video- und Tonsequenzen beinhalten, spielt in multimedialen Informationssystemen eine zentrale Rolle. Die Anwendbarkeit dieser Systeme hängt sehr stark davon ab, inwieweit der Zugriff auf diese Daten sowie deren effiziente Erschließung und Indexierung unterstützt wird. Ein Thema der Vorlesung sind daher neue Ansätze aus dem Bereich des Information Retrieval, die einen inhalts-orientierten, struktur-orientierten oder Kontext/Meta-Daten-orientierten Zugriff auf Multimedia-Dokumente ermöglichen. Ein weiteres Themengebiet sind neue Konzeptionen für Multimedia-Informationen-Retrieval-Programmier- und Benutzungsschnittstellen. Diese haben insbesondere im Bereich der Multimedia-Informationssysteme eine hohe Relevanz, da konventionelle Anfragesprachen und die darauf basierenden Interfaces auf die Erfordernisse und Möglichkeiten des bislang vorherrschenden textorientierten Information Retrieval zugeschnitten sind. Ziel der Vorlesung ist es, einen einführenden Überblick über aktuelle Technologien sowie zum gegenwärtigen Stand der Forschung und Entwicklung in den relevanten Themengebieten Multimediatechnologien sowie Hypermedia- und Multimedia-Information-Retrieval- Methoden und -Technologien zu geben.</p> <p>Kurs 01876: Die Archivierung von Multimediadaten soll deren Langzeitverfügbarkeit gewährleisten, d. h. die volle Originalität und Funktionalität eines digitalen Objekts auch für eine zukünftige Nutzung garantieren. Die Vorlesung befasst sich neben der Vorstellung einschlägig relevanter Multimedia-Storage, -Server, -Content Management-Lösungen aus der produktiven Praxis auch mit teilweise bereits im Einsatz oder gerade in der Entstehung befindlichen Technologien und Systemen sowie notwendigen technischen Formaten und Normen zur Berücksichtigung des technologischen Wandels innerhalb der Archivierungsdauer. Welche Strategien sollten eingeschlagen werden, um multimediale Datenbestände mit vertretbarem Aufwand langfristig zugänglich und nutzbar halten zu können? Hierbei wird zwischen verschiedenen Anwendungen aus dem klassischen Bibliothekswesen, aber insbesondere auch auf die speziellen</p>		

Anforderungen von Broadcast-, Produktions- und Re-Use-Szenarien eingegangen. Die in der Vorlesung behandelten Aspekte umfassen Technologien für Digitale Bibliotheken, Aufbau und Struktur multimedialer Datenobjekte und korrespondierender Metadatenschemata. Weiterhin werden Techniken für die Segmentierung, Verfahren für die Sicherung von Authentizität und Integrität, Methoden für die Extraktion von semantischen Informationen sowie der Repräsentation komplexer Verknüpfungen diskutiert. Weiterhin werden die aus informatischer Sicht relevanten rechtlichen Regelungen und urheberrechtlichen Beschränkungen skizziert.

Inhaltliche Voraussetzung	Kenntnisse aus dem Modul 63413 "Daten- und Dokumentenmanagement im Internet" (01877)	
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende	
Anmerkung	Das Modul Multimedialinformationssysteme kann nicht mehr neu belegt werden. Nach grundlegender Überarbeitung wird das Modul voraussichtlich wieder ab Wintersemester 2022/23 belegbar sein. Eine Prüfungsteilnahme ist weiterhin möglich.	
Formale Voraussetzung	keine	
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik M.Sc. Wirtschaftsinformatik	
<b>Prüfungsformen</b>	Art der Prüfungsleistung	Voraussetzung
Prüfung	bestandene benotete Prüfungsklausur	keine
Stellenwert der Note	1/12	

Lehrende/r	Felix Engel	Modulbeauftragte/r	Matthias Hemmje
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01879 Information Retrieval		WS SWS 4+2
Detaillierter Zeitaufwand	Bearbeitung der Kurseinheiten (7 mal 20 Stunden): 140 Stunden Einüben der Inhalte (7 mal 15 Stunden): 105 Stunden Wiederholung und Prüfungsvorbereitung: 55 Stunden		
Qualifikationsziele	Die Studierenden beherrschen die wichtigsten allgemeinen Begriffe, Modelle und Methoden des Information Retrieval und können diese innerhalb exemplarischer Technologien und Anwendungen zuordnen sowie innerhalb eigener Entwürfe und Modellierungen von Information Retrieval Anwendungen in Benutzungsschnittstellen zu Informationssystemen anwenden.		
Inhalte	<p>Beständig- und im zunehmend Maße werden Informationen direkt in digitaler Form erzeugt oder nachträglich in ein digitales Format überführt. Ein Grund dafür ist die schnelle und einfache Verarbeitung und eine damit einhergehende bessere Wiederverwendbarkeit. Einen umfangreichen digitalen Datenbestand jedoch manuell und gezielt nach einer bestimmten Information zu durchsuchen ist ab einer bestimmten Menge an Daten nicht mehr effektiv möglich und der tatsächliche Nutzen des Bestands damit zumindest fraglich. Ein plakatives Beispiel für einen multimedialen Datenbestand ist das Internet, welches massive Mengen an digitalen Daten vorhält. Wohlbekannte Suchmaschinen helfen hier dem suchenden, um sich in diesem Bestand zurechtzufinden. Große Datenbestände entstehen jedoch auch in spezielleren Bereichen, wie z.B. in Behörden, Krankenhäusern oder Verlagen. Auch hier muss ein effektives Auffinden gesuchter Informationen gewährleistet werden. Die Forschung im Umfeld des Information Retrieval (IR) befasst sich daher mit der Modellierung und Umsetzung von Anwendungen die automatisiert digitale Datenbestände, für den einfachen Zugriff und Nachnutzung aufbereiten. Die Forschung an effektiven IR-Verfahren ist hinreichend komplex und obwohl das IR auf eine lange Historie zurückblick sind insbesondere mit Hinblick auf anwachsende Datenmengen, mit zunehmend heterogener und verteilter Natur, Fragestellungen offengeblieben und neue Anforderungen hinzugekommen.</p> <p>Dieser Kurs wird sich zunächst mit klassischen Themen des IR in Dokumentdatenbeständen befassen, um die Grundlegendem Eigenschaften einzuführen. Darunter fallen Themen wie die Indexierung von Text und Verfahren zu Gewichtungen von Indexeinheiten, die Einführung etablierter IR-Klassen und Modelle, sowie Verfahren zu Evaluation von IR-Verfahren. Über die klassischen IR-Verfahren hinaus wird sich dieser Kurs dann noch den Themen der semantischen Suche und der verteilten Suche widmen.</p>		
Inhaltliche Voraussetzung	Keine		
Lehr- und Betreuungsformen	Kursmaterial internetgestütztes Diskussionsforum Einsendeaufgaben mit Korrektur und/oder Musterlösung		
Anmerkung	Das Modul kann letztmalig im WS 2022/23 belegt werden. Nach grundlegender Überarbeitung wird das Modul wieder belegbar sein. Eine Prüfungsteilnahme ist weiterhin möglich.		

Formale Voraussetzung keine

Verwendung des Moduls M.Sc. Informatik  
M.Sc. Praktische Informatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete Prüfungsklausur	keine
Stellenwert der Note	1/12		

## Lehrende/r

Tobias Vogel

Modulbeauftragte/r

Matthias Hemmje

Dauer des Moduls  
ein SemesterECTS  
10Workload  
300 StundenHäufigkeit  
in jedem Semester

## Lehrveranstaltung(en)

01774 Intelligente Informationssysteme für industrielle Anwendungen WS/SS SWS  
4+2

## Detaillierter Zeitaufwand

Bearbeitung der Kurseinheiten (7 mal 20 Stunden): 140 Stunden  
Einüben der Inhalte (7 mal 15 Stunden): 105 Stunden  
Wiederholung und Prüfungsvorbereitung: 55 Stunden

## Qualifikationsziele

Das übergeordnete Ziel dieses Kurses ist der Erwerb von Kompetenz und Handlungsfähigkeit zu intelligenten Informationssystemen in industriellen Anwendungen. Dazu wird ein grundlegendes Verständnis zu Daten, Information und Informationssystemen vorgestellt, mit dem Ziel, dass die Kursteilnehmerinnen und Kursteilnehmer die Definitionen, Ansätze, Methoden und Modelle verstehen und strukturiert wiedergeben können.

Darauf aufbauend eignen sich die Teilnehmerinnen und Teilnehmer erweiterte Kenntnisse zu aktuellen (semantischen) Technologien des Informations-, Ressourcen- und Prozessmanagements sowie zu wissensbasierten und intelligenten Informationssystemen an. Dazu gehört, dass die Kursteilnehmerinnen und Kursteilnehmer den sicheren Umgang mit der Fachterminologie zu wissensbasierten und intelligenten Informationssystemen beherrschen sowie ein eigenes Verständnis zur Einordnung von Semantik mit den Basiskonzepten, den grundlegenden semantischen Modellen und zu Ontologien aufbauen. Weiterhin erkennen die Kursteilnehmerinnen und Kursteilnehmer die Eigenschaften und Abgrenzung der intelligenten Informationssysteme. Die Studierenden sind nach Abschluss des Kurses in der Lage, wissensbasierte Prozessrepräsentationen, (intelligente) Informationssysteme und industrielle Anwendungen zu beschreiben und zu beurteilen. Sie können die wichtigsten Begriffe, Modelle und Methoden zu intelligenten Informationssystemen im Kontext der industriellen Anwendung aufzeigen und formulieren.

Die Kursteilnehmerinnen und Kursteilnehmer vertiefen und reflektieren das Erlernte zudem anhand der Selbsttestaufgaben abschließend zu jeder Kurseinheit.

## Inhalte

In diesem Kurs 01774, bestehend aus sieben Kurseinheiten, betrachten die Teilnehmerinnen und Teilnehmer zunächst intelligente Informationssysteme. Sie erlernen die wesentlichen Begriffe zu Daten, Information und Wissen sowie zur Semantik und der semantischen Repräsentation von Wissen und Ressourcen im Zusammenhang mit industriellen Wertschöpfungsprozessen.

Aufbauend auf den Grundlagen der Industrie 4.0 erarbeiten die Studierenden die historisch verfügbaren Informations-, Planungs- und Steuerungssysteme, wie z.B. Enterprise Resource Planning (ERP) Systeme im industriellen Einsatzgebiet. Im Umfeld der industriellen Wertschöpfungsprozesse wird ein semantisches Prozessmodell zu den betrachteten Wissensbasierten Prozess Repräsentationen (engl. Knowledge-based [x] Prozess Representation, KxP) vorgestellt und durch die Teilnehmerinnen und Teilnehmer erarbeitet. Dabei identifizieren die Teilnehmerinnen und Teilnehmer die zentralen Prozessphasen dieser wissensbasierten Prozesse, beginnend mit der semantischen Prozessbeschreibung, der dreistufigen Planungs- und Mediationsphase sowie der Prozessdokumentation und abschließenden Archivierung. Dieses semantische Prozessmodell zu den Knowledge-based [x] Processes (KxP) wenden die Studierenden auf verschiedene [x]-Bereiche der Wertschöpfung an, z.B. auf Wissensbasierte Innovations Prozesse (engl. Knowledge-based Innovation Processes, KIP), Wissensbasierte Entwicklungs Prozesse (engl. Knowledge-based Engineering Processes, KEP) und Wissensbasierte Produktions Prozesse (engl. Knowledge-based Production Processes, KPP). Die Prozessrepräsentationen zu den KxP Prozessen werden im sog.

Wissensbasierten Prozesslebenszyklus Management (engl. Knowledge-based Process Lifecycle Management, KPLM) zusammengefasst. Zudem analysieren und hinterfragen die Teilnehmerinnen und Teilnehmer, welchen Nutzen, welche Analysemöglichkeiten und welche Flexibilisierung ein semantischer, digitaler Zwilling zu Wertschöpfungsprozessen, der sog. Semantische Process Twin (SPT) für die Industrie 4.0 erzielen kann.

Inhaltliche  
Voraussetzung

Keine

Lehr- und  
Betreuungsformen

Kursmaterial  
Betreuung und Beratung durch Lehrende  
Einsendeaufgaben mit Korrektur und/oder Musterlösung

Anmerkung

Keine

Formale Voraussetzung

keine

Verwendung des Moduls

M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

## Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete Prüfungsklausur

Die Zulassung zur Klausur kann nur erfolgen, wenn bei 5 von 7 Kurseinheiten mindestens 50 % der möglichen Punkte in den Einsendeaufgaben erreicht wurden.

Stellenwert  
der Note 1/12

Lehrende/r	Kai Sauerwald	Modulbeauftragte/r	Kai Sauerwald
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01695 Deduktions- und Inferenzsysteme		SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 130 - 150 Stunden, bearbeiten der Übungsaufgaben: 60 - 75 Stunden, Studientage und Prüfungsvorbereitung: 60 - 75 Stunden		
Qualifikationsziele	Die Studierenden können grundlegenden Begriffe, Methoden und Verfahren der Deduktions- und Inferenzsysteme präsentieren, die entsprechenden formalen Grundlagen beschreiben und zentrale Verfahren wie z.B. das Resolutionsverfahren anwenden.		
Inhalte	Intelligentes Verhalten basiert wesentlich auf der Fähigkeit, logische Schlüsse zu ziehen, und in nahezu allen Systemen der künstlichen Intelligenz spielen automatische Inferenz- oder Deduktionskomponenten eine zentrale Rolle. Anwendungsfelder sind etwa das automatische Beweisen mathematischer Sätze, logische Programmiersprachen (z. B. PROLOG), Programmverifikation, deduktive Planverfahren oder die Entwicklung von Inferenzkomponenten für spezifische Wissensrepräsentationssprachen. In diesem Kurs werden grundlegende Begriffe, Methoden und Verfahren der Deduktions- und Inferenzsysteme behandelt: Kalküle für die Prädikatenlogik 1. Stufe, Resolutionsverfahren, Repräsentation von Kalkülen, Gleichheit und Unifikation, Termersetzungssysteme, Deduktion und logisches Programmieren, nicht- klassische Formen der Inferenz.		
	Ergänzende Literatur: K. H. Bläsius, H.J. Bürckert (Hrsg.): Deduktionssysteme. Automatisierung des logischen Denkens. Oldenbourg-Verlag, 1992. W. Bibel, S. Hölldobler, T. Schaub: Wissensrepräsentation und Inferenz. Vieweg-Verlag, 1993.		
Inhaltliche Voraussetzung	keine		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum		
Anmerkung	Ab Wintersemester 2022/2023 wird das Modul regelmäßig in jedem Wintersemester angeboten.		
Formale Voraussetzung	keine		
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik M.Sc. Wirtschaftsinformatik		



## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

bestandene benotete mündliche  
Modulprüfung

Voraussetzung

keine

Lehrende/r	Matthias Thimm	Modulbeauftragte/r	Matthias Thimm
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01845	Methoden der Wissensrepräsentation und -verarbeitung	WS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 130 - 150 Stunden, Bearbeiten der Übungsaufgaben: 60 - 75 Stunden, Studientage und Prüfungsvorbereitung: 60 - 75 Stunden		
Qualifikationsziele	Die Studierenden können fortgeschrittene Methoden zur Repräsentation und Verarbeitung unsicheren und vagen Wissens, insbesondere aus dem Bereich des revidierbaren Schließens und der quantitativen Methoden wiedergeben und die Grundlagen des Bereichs Aktionen und Planen beschreiben. Sie können Vor- und Nachteile der unterschiedlichen Ansätze erklären und diese Ansätze in entsprechenden Problemstellungen anwenden.		
Inhalte	Für die Realisierung maschineller Intelligenz ist die Frage der Wissensrepräsentation und -verarbeitung von zentraler Bedeutung. Jedes wissensbasierte System kombiniert das in seiner Wissensbasis gespeicherte Wissen mit aktuellen Informationen und kommt so zu "Erkenntnissen", die es beispielsweise dem Benutzer in Form von Diagnosen präsentiert, oder die es selbst - als autonom agierendes System - als Grundlage seiner Handlungen nutzt. Häufig ist dabei das zu verarbeitende Wissen unsicherer oder unvollständiger Natur, so dass Methoden zum Einsatz kommen müssen, die auch unter diesen Umständen vernünftige Resultate liefern. Dies macht die Behandlung von Inkonsistenzen erforderlich und führt auf die Problematik des nichtmonotonen oder revidierbaren Schließens. Einen anderen Ansatz zur Modellierung von Unsicherheit bieten die quantitativen Methoden. Ferner gehen wir auf den Bereich Aktionen und Planen ein, in dem sich die Veränderung von Wissen durch aktive Eingriffe (Handlungen) als eine Kernproblematik erweist. Der Kurs behandelt schwerpunktmäßig die folgenden Themen: Nichtmonotones Schließen, Truth Maintenance-Systeme, Default-Logiken, Aktionen und Planen, Situationskalkül, Wahrscheinlichkeit und Information, probabilistische Netzwerke, insbesondere Markov- und Bayes-Netze, Fuzzy- und Dempster-Shafer-Theorie, zahlreiche Anwendungsbeispiele, u.a. aus Technik, Medizin und Genetik.		
Inhaltliche Voraussetzung	Kenntnisse formaler Grundlagen der Informatik; hilfreich sind auch Grundkenntnisse im Bereich der Logik und aus dem Gebiet der wissensbasierten Systeme, z.B. Modul 64211 "Wissensbasierte Systeme" (01696).		
Lehr- und Betreuungsformen	Kursmaterial Einsendaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum		
Anmerkung			
Formale Voraussetzung	keine		
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik M.Sc. Wirtschaftsinformatik		

## Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete Prüfungsklausur

keine

Stellenwert  
der Note 1/12

# Katalog M4: Softwaretechnik und Programmierung

Lehrende/r	Friedrich Steimann	Modulbeauftragte/r	Friedrich Steimann
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01814	Objektorientierte Programmierung	SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeitung des Kurstextes und Einarbeitung in Smalltalk: 200 Stunden Bearbeitung der Übungs- und Einsendeaufgaben: 50 Stunden Nachbearbeitung und Klausur- bzw. Prüfungsvorbereitung: 50 Stunden		
Qualifikationsziele	Die Studierenden entwickeln ein Verständnis für objektorientiertes Denken und Programmieren. Sie kennen die Konstrukte objektorientierter Programmiersprachen und sind in der Lage, dabei insbesondere die Bedeutung eines Typsystems richtig einzuschätzen. Sie können die Schwächen der objektorientierten Programmierung benennen, Kriterien für die Auswahl einer geeigneten Programmiersprache aufstellen und Aussagen zum eigenen und zum Programmierstil anderer machen.		
Inhalte	<p>Objektorientierung ist ein weit verbreiteter Standard der Programmierung. Dieser Kurs soll die Voraussetzungen für die Einordnung und das schnelle Beherrschen verschiedenster objektorientierter Programmiersprachen schaffen. Zugleich soll er den Studierenden Kriterien an die Hand geben, mithilfe derer sie die Eignung bestimmter objektorientierter Programmiersprachen für vorgegebene Zwecke beurteilen können.</p> <p>Der Kurs führt die objektorientierte Programmierung anhand der Programmiersprache Smalltalk ein. Smalltalk zeichnet sich nicht nur durch eine besonders konsequente Umsetzung objektorientierter Konzepte aus, sondern stellt auch die Verbindung zur funktionalen Programmierung her, die für die heutige objektorientierte Programmierung stilprägend ist. Smalltalk ist schnell und leicht erlernbar, dies nicht zuletzt auch deswegen, weil es kein statisches Typsystem besitzt. Da statische Typsysteme aber bereits vor der Ausführung von Programmen logische Fehler aufzudecken erlauben, müssen sie für die kommerzielle Softwareentwicklung als unverzichtbar angesehen werden. Der Kurs widmet sich daher in der Folge ausschließlich solchen Programmiersprachen, die über eine statische Typprüfung verfügen, darunter Java, C# und C++. Abgerundet wird der Kurs durch die Behandlung der häufig ignorierten, aber dennoch nicht zu vernachlässigenden Probleme der objektorientierten Programmierung sowie durch eine Darstellung objektorientierten Programmierstils.</p> <p><b>Ergänzende Literatur:</b></p> <p>A. Goldberg, D. Robson Smalltalk-80: The Language and Its Implementation (Addison-Wesley 1983).</p> <p>B. Meyer, Object-Oriented Software Construction 2. Ausgabe (Prentice Hall, 2000).</p>		
Inhaltliche Voraussetzung	Der Kurs richtet sich an Interessierte in fortgeschrittenen Studienabschnitten. Belegung der Module 63811 "Einführung in die imperative Programmierung" (01613) und 63611 "Einführung in die objektorientierte Programmierung" (01618) ist hilfreich, aber keine notwendige Bedingung.		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende		
Anmerkung	Achtung: Dieser Kurs ist kein Java-Kurs. Java wird in 01618 in großer Vollständigkeit behandelt. Auch befasst sich dieser Kurs nicht mit Scripting-Sprachen. Einsende- und		

Selbsttestaufgaben verlangen die Bereitschaft, in Smalltalk zu programmieren. Ihre Bearbeitung wird dringend empfohlen.

Formale Voraussetzung keine

Verwendung des Moduls M.Sc. Informatik  
M.Sc. Praktische Informatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/12	Modulprüfung	

Lehrende/r	Daniela Keller	Modulbeauftragte/r	Daniela Keller
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01853	Moderne Programmier Techniken und -methoden	WS/SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeitung des Kurstextes: 150 Stunden Bearbeitung der Übungs- und Einsendeaufgaben: 75 Stunden Nachbearbeitung und Klausur- bzw. Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	Die Studierenden verstehen die Ideen der interfacebasierter Programmierung und sind in der Lage, die interfacebasierte Programmierung anzuwenden. Sie wissen, was man unter Design by contract versteht und warum sie es anwenden sollten. Sie kennen die Vor- und Nachteile verschiedener Sprachen zur Umsetzung von Design by Contract. Sie wissen um die Nützlichkeit von Unit-Testen und wie man das Framework JUNIT anwendet. Sie kennen die wichtigsten Entwurfsmuster und können sie bei der Software-Entwicklung verwenden. Sie wissen, was man unter Refactoring versteht und kennen verschiedene Refaktorisierungswerkzeuge und die Vorbedingungen für ihre Anwendbarkeit. Sie wissen, was man unter Metaprogrammierung versteht und können Vor- und Nachteile der Metaprogrammierung beurteilen. Sie wissen, wie die agilen Methoden wie das Extreme Programming funktionieren und welche Lösungen sie für die Probleme der Software-Entwicklung bieten.		
Inhalte	Trotz eines zunehmenden Bewusstseins für die Notwendigkeit von vorbereitenden Tätigkeiten wie etwa der Softwaremodellierung ist die Programmierung immer noch die Kernaktivität der Softwareentwicklung. Zwar hat die Einführung neuer Programmiersprachen wie Java und C# durchaus Produktivitätssteigerungen bewirkt, jedoch ist die Fortentwicklung dieser Sprachen viel zu schwerfällig, um mit den ständig wachsenden Ansprüchen an Funktionalität und Umfang schritthalten zu können. Statt dessen haben sich im Kontext der objektorientierten Programmierung eine ganze Reihe von Techniken und Methoden entwickelt, mit deren Hilfe sich - auf der Basis existierender Pro-grammiersprachen - die Softwareentwicklung effizienter gestalten lässt. Einige dieser Programmier Techniken und -methoden werden in diesem Kurs vorgestellt.		
	<b>Ergänzende Literatur:</b> E. Gamma et al. Design Patterns (Addison-Wesley 1995). M. Fowler: Refactoring: Improving the Design of Existing Code (Addison-Wesley 1999), Refactoring (deutsche Version) (mitp Professional 31.10.2019) K. Beck, C. Andres: Extreme Programming Explained. Embrace Change 2. Ausgabe (Addison Wesley 2004).		
Inhaltliche Voraussetzung	Es werden Programmierkenntnisse in Java vorausgesetzt, wie sie z. B. durch das Modul 63611 "Einführung in die objektorientierte Programmierung" (01618) vermittelt werden. Zusätzlich ist ein durch Praxis erworbenes Gefühl für die objektorientierte Programmierung Voraussetzung, um die Bedeutung der im Kurs geschilderten Probleme und Lösungen abschätzen zu können.		
Lehr- und Betreuungsformen	Kursmaterial internetgestütztes Diskussionsforum Einsendeaufgaben mit Korrektur und/oder Musterlösung		
Anmerkung	-		
Formale Voraussetzung	keine		

Verwendung des Moduls M.Sc. Informatik  
M.Sc. Praktische Informatik  
M.Sc. Wirtschaftsinformatik

**Prüfungsformen**

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete mündliche  
Modulprüfung

keine

Stellenwert 1/12  
der Note



# Fachpraktika

Lehrende/r	Jörg M. Haake	Modulbeauftragte/r	Jörg M. Haake
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01592 Fachpraktikum CSCW		WS SWS 4
Detaillierter Zeitaufwand	Bearbeiten des Gruppenprojekts: 260 Stunden Teilnahme an beiden Präsenzphasen: 40 Stunden		
Qualifikationsziele	Die Studierenden entwickeln ein tiefergehendes Verständnis für die Probleme und Lösungsalternativen bei der Realisierung von Groupware-Anwendungen und/oder den zugrundeliegenden Frameworks/Toolkits. Sie können eine Groupware-Anwendung spezifizieren, entwerfen und in einer modernen Programmierumgebung realisieren, testen und dokumentieren. Sie besitzen die Kompetenz zu verteilter Teamarbeit, insbesondere zur Wahrnehmung von Rollen im Projektmanagement, bei der Nutzung verteilter Entwicklungsumgebungen sowie in gemeinsamen Arbeitsumgebungen. Sie beherrschen den Einsatz von Versionierungssystemen und Groupware-Werkzeugen zur Unterstützung der Arbeit eines verteilten Projektteams.		
Inhalte	Dieses Fachpraktikum behandelt Design und Implementierung von CSCW-Systemen (Groupware). Die Teilnehmenden entwickeln in einer Projektgruppe ein konkretes CSCW-System. Insbesondere werden Methoden für die Anforderungsermittlung, den Entwurf, die Realisierung, den Test und die Dokumentation von Groupware-Anwendungen (d.h. von Anwendungen für die Unterstützung von Gruppenarbeit, CSCW) erlernt und in der Gruppe eingeübt. Neben diesen technischen Aspekten der Softwareentwicklung werden Methoden für die Organisation der Projektarbeit ("Software Engineering in the large") in einem verteilten Team behandelt. Im Fachpraktikum nutzen die Projektgruppen moderne Entwicklungsumgebungen, Groupware-Werkzeuge und Versionierungssysteme. Das Projektmanagement zur Bearbeitung der Entwicklungsaufgabe wird vom Team durchgeführt. Die Teilnehmenden des Praktikums erhalten die Möglichkeit, während des Praktikums verschiedene Rollen aus der Projektarbeit einzuüben. Jedes Team stellt seine Ergebnisse in einer Abschlusspräsentation vor.		
	<b>Ergänzende Literatur:</b> Literatur zu den genutzten Programmiersprachen und ggf. Frameworks wird zum Beginn des Praktikums bekannt gegeben.		
Inhaltliche Voraussetzung	Für die Teilnahme am Fachpraktikum sind fundierte Kenntnisse der objektorientierten Software-Entwicklung (vor allem Design und Implementierung) erforderlich. Hilfreich sind fundierte Kenntnisse in Verteilten Systemen und Kooperativen Systemen, wie sie in den Modulen 63211 "Verteilte Systeme" (01678), 63214 "Computerunterstütztes kooperatives Arbeiten und Lernen" (01880/01883) oder 63215 "Gestaltung kooperativer Systeme" (01884) erworben werden können.		
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende Zusatzmaterial		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		
Verwendung des Moduls	B.Sc. Informatik		

M.Sc. Informatik  
M.Sc. Praktische Informatik

**Prüfungsformen**

Prüfung  
Stellenwert 1/12  
der Note

Art der Prüfungsleistung  
erfolgreich bearbeitete  
Praktikumsaufgabe

Voraussetzung  
keine

Lehrende/r	Gabriele Peters	Modulbeauftragte/r	Gabriele Peters
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit alle 2 bis 3 Jahre
Lehrveranstaltung(en)	01513 Fachpraktikum Mensch-Computer-Interaktion		SWS 4
Detaillierter Zeitaufwand	Bearbeitung des Projektes: 250 Stunden Präsentation der Ergebnisse: 50 Stunden		
Qualifikationsziele	<p>Erfolgreiche Absolventinnen und Absolventen verfügen nach dem Praktikum über Kenntnisse des aktuellen Forschungsstands eines Gebietes der Entwicklung interaktiver und intelligenter Systeme. Sie haben die Fähigkeit erworben, auf der Grundlage von wissenschaftlicher Originalliteratur eigene Lösungskonzepte für aktuelle Forschungsfragen der Mensch-Computer-Interaktion zu entwickeln. Im Rahmen der Softwareentwicklung in Teamarbeit haben die Absolventinnen und Absolventen eine koordinierte Arbeitsteilung zur gemeinsamen Lösung eines komplexen Problems erlernt. Dies umfasst die Ausbildung bestimmter Rollen im Projektmanagement (z.B. Projektleitung), die Verwendung von kollaborativen Entwicklungswerkzeugen (z.B. git) und eine fachgerechte Dokumentation der Komponenten des implementierten Systems. Darüber hinaus sind sie in der Lage, ihre Arbeiten in einer Abschlussveranstaltung zu präsentieren und zu vertreten.</p>		
Inhalte	<p>Im Fachpraktikum des Lehrgebiets Mensch-Computer-Interaktion werden aktuelle Probleme aus dem Bereich der Entwicklung interaktiver und intelligenter Systeme bearbeitet. Dazu gehören u.a. die Verarbeitung komplexer Sensorsignale wie etwa Video-, Audio- oder 3D-Daten sowie die Anwendung von Methoden des maschinellen Lernens (z.B. Reinforcement Learning). Auch Verfahren des maschinellen Sehens oder der algorithmischen Generierung von Inhalten können zum Aufgabenspektrum des Fachpraktikums gehören. Darüber hinaus können je nach Aufgabenstellung auch Systeme des Interaktionslabors in Hagen eingesetzt werden, für deren Verwendung die Studierenden ihre Lösungsansätze zunächst in Simulationen testen, bevor sie in der Abschlussphase des Fachpraktikums auf den realen Systemen zum Einsatz kommen. Das entwickelte Softwaresystem wird in einer Abschlussveranstaltung präsentiert.</p>		
Inhaltliche Voraussetzung	<p>Die in dem Modul 63312 "Interaktive Systeme" (01698/01699) erworbenen oder äquivalente Kenntnisse sind wünschenswert. Des Weiteren sollen die Teilnehmer und Teilnehmerinnen über praktische Programmiererfahrung in den Sprachen C/C++ oder Java verfügen.</p>		
Lehr- und Betreuungsformen	<p>internetgestütztes Diskussionsforum Zusatzmaterial Betreuung und Beratung durch Lehrende</p>		
Anmerkung	<p>Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a>.</p>		
Formale Voraussetzung	<p>mindestens zwei Wahlmodulprüfungen müssen bestanden sein</p>		
Verwendung des Moduls	<p>B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik</p>		

**Prüfungsformen**

Prüfung  
Stellenwert  
der Note 1/12

Art der Prüfungsleistung

erfolgreich bearbeitete  
Praktikumsaufgabe

Voraussetzung

keine

Lehrende/r	Dominic Heutelbeck	Modulbeauftragte/r	Dominic Heutelbeck
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01598 Fachpraktikum für sichere kollaborative Anwendungen		WS SWS 4
Detaillierter Zeitaufwand	Präsenzphase: 70 Stunden Heimarbeit: 230 Stunden		
Qualifikationsziele	Nach erfolgreicher Teilnahme am Fachpraktikum können Studierende die behandelten Entwurfsmuster und Technologien praktisch anwenden. Sie haben Erfahrungen in Teamarbeit und Aufgabenorganisation erworben. Sie wissen außerdem, wie man Ziele eigenständig definiert und wie diese durch Projekthandbücher durchgesetzt werden. Sie haben Erfahrungen mit dem Einsatz von Projektmanagementprogrammen und sie können mit einem Versionskontrollsystem umgehen.		
Inhalte	<p>Im Fachpraktikum für sichere kollaborative Anwendungen bieten wir den Studierenden die Möglichkeit, im Team ein komplexes Softwareprodukt zu erstellen. Dabei erlernen die Studierenden moderne Entwurfsmuster und aktuelle Technologien. Das Anwendungsszenario dieses Praktikums sind sichere Dienste für die fertigende Industrie erstellt.</p> <p>In Abhängigkeit von der Teilnehmerzusammensetzung können dabei folgende Bereiche bearbeitet werden:</p> <ul style="list-style-type: none"> <li>- Policy und Datenstrom basierte Autorisierung von Datenzugriffen.</li> <li>- Kollaborative Benutzungsschnittstellen. Web Basiert (Vaadin, Java, Spring, Python, Django).</li> <li>- Smart Contracts und Blockchain (z. B. LTO Network, Ethereum)</li> <li>- Internet der Dinge: Sensordaten von Maschinen (z. B. OPC-UA, MQTT)</li> <li>- Backend Infrastrukturen mit Microservice-Architekturen, Domain Driven Design, CQRS-ES, Java, Spring und Axon.</li> </ul> <p>Die Bearbeitung der gestellten Aufgaben erfolgt in Kleingruppen. Die Natur der gestellten Aufgaben setzt einen entsprechend ausgestatteten Rechner mit Internetzugang voraus. Die Kommunikation mit den Teilnehmern über verschiedene Onlineplattformen und mit E-Mail ist dabei unabdingbar.</p>		
Inhaltliche Voraussetzung	Das Fachpraktikum richtet sich primär an fortgeschrittene Studierende. Voraussetzung sind gute Kenntnisse im Bereich Software Engineering und Java oder mit Smart Contracts. Das Praktikum macht starken Gebrauch von fortgeschrittenen Entwurfsmustern. Daher wird einschlägiges Vorwissen vorausgesetzt.		
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik		

## M.Sc. Praktische Informatik

### Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

erfolgreich bearbeitete  
Praktikumsaufgabe

Voraussetzung

keine

Lehrende/r	Matthias Hemmje	Modulbeauftragte/r	Matthias Hemmje
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01594 Fachpraktikum Multimedia- und Internetanwendungen		SWS 4
Detaillierter Zeitaufwand	Präsenzphasen: 50 Stunden Präsentationsvorbereitung: 10 Stunden Entwurf und Implementierung von Algorithmen im Team: 240 Stunden		
Qualifikationsziele	Nach erfolgreicher Teilnahme am Fachpraktikum beherrschen Studierende den praktischen Umgang mit neuesten Multimedia- und Internettechnologien und sind in der Lage, die Erfahrungen und Kompetenzen direkt in einer technisch-wissenschaftlichen Softwareentwicklung einzusetzen. Zudem erwerben die Teilnehmerinnen und Teilnehmer Erfahrungen und Kompetenzen in Teamarbeit und Aufgabenorganisation im Team. Sie wissen, wie man Ziele eigenständig definiert und wie diese durch kollaborative Arbeits- und Zeitplanung erreicht werden. Sie können die geplanten Arbeiten in einem Exposé beschreiben und die Ergebnisse in einer Abschlussdokumentation vorstellen und vertreten. Sie können schriftliche Planungen und Dokumentationen sowie implementierte Module in einem Versionierungssystem für die Softwareentwicklung ablegen.		
Inhalte	Die Forschung und Lehre des Lehrgebietes bewegt sich in den Bereichen Daten- und Dokumentenmanagement im Internet, Informations- und Wissensmanagement im Internet, Multimedia-Informationssysteme und Datenbanken, Informationsvisualisierung im Internet. Dazu gehören des Weiteren die Forschung, Lehre und Entwicklung in den folgenden Bereichen der Informatik und ihren Anwendungsgebieten: Informationsvisualisierung im Internet, Information Retrieval, Visuelle Mensch-Maschine-Interaktion, Content- und Wissensmanagement im Internet, Semantic Web, Digitale Langzeitarchivierung, Virtuelle Forschungsumgebungen, Big Data Analyse, Analyse natürlicher Sprache, Berufliche Weiterbildung und E-Learning, Industrie 4.0 und „Factories of the Future“. Neben der Kooperation mit nationalen und internationalen Forschungs-/Entwicklungs- und industriellen Endanwendungspartnern unterstützt das Lehrgebiet in Kooperation mit dem An-Institut FTK, Forschungsinstitut für Telekommunikation und Kooperation in Dortmund, auch den Transfer der Forschungsergebnisse in innovative Prototypen, Produkte und Dienste. Die Teilnehmerinnen und Teilnehmer des Praktikums dürfen zunächst aus einer Menge an Aufgabenstellungen, gemäß ihres Interesses eine Auswahl für die weitere Bearbeitung treffen. Die Bearbeitung der gestellten Aufgabe erfolgt dann modular in kooperierenden Kleingruppen.		
Inhaltliche Voraussetzung	Gute Kenntnisse in JAVA und objektorientierter Softwareentwicklung werden vorausgesetzt. Der Umgang mit Technologien wie Eclipse, Maven und GIT sollte bekannt sein.		
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		



Verwendung des Moduls B.Sc. Informatik  
M.Sc. Informatik  
M.Sc. Praktische Informatik

**Prüfungsformen**

Art der Prüfungsleistung

Voraussetzung

Prüfung

erfolgreich bearbeitete

keine

Stellenwert 1/12  
der Note

Praktikumsaufgabe

Lehrende/r	Jörg Keller	Modulbeauftragte/r	Jörg Keller
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit 1 x jährlich
Lehrveranstaltung(en)	01599	Fachpraktikum IT-Sicherheit	SWS 4
Detaillierter Zeitaufwand	Bearbeitung Aufgaben Phase 1: 150 Stunden Bearbeitung Aufgaben Phase 2: 100 Stunden Dokumentation u. Präsentation: 50 Stunden		
Qualifikationsziele	Nach erfolgreicher Bearbeitung können Studierende die üblichen Werkzeuge der IT-Sicherheit bedienen und konfigurieren. Sie sind in der Lage, die Grundlagen der Funktionsfähigkeit von Firewalls nach Änderungen der Konfiguration zu überprüfen. Sie kennen die relevanten Log-Dateien und können die Bedeutung von Einträgen interpretieren. Die Teilnehmerinnen und Teilnehmer sind in der Lage, sich in einem Team zu organisieren, effizient an der Lösung einer Aufgabe zu arbeiten, und die dabei auftretenden Differenzen einer Lösung zuzuführen. Die Teilnehmerinnen und Teilnehmer können ihre Entscheidungen und Maßnahmen bei Administration und Installation von einigen Werkzeugen der IT-Sicherheit begründen und präsentieren.		
Inhalte	In diesem Fachpraktikum sollen die in den Kursen Sicherheit im Internet I+II vermittelten Kenntnisse anhand praktischer Aufgabenstellungen angewendet werden. Jede/r Praktikumssteilnehmerin/-teilnehmer erhält Zugang zu einem zentralen Übungsrechner, auf dem er einen eigenen virtuellen Linux-Rechner schützen soll. Hierzu sind ein gesicherter Zugang (VPN), eine Firewall, ein Application-Level Gateway, ein Intrusion Detection System und weitere Werkzeuge zur Sicherung zu installieren, zu konfigurieren und zu testen. Anschließend wird in Gruppen eine größere Aufgabenstellung wie die Installation und Konfigurationen eines VPN zwischen mehreren Gruppen kollaborativ gelöst. Neben diesen Arbeiten obliegt auch die Organisation des Managements dieses Projekts den Studierenden. Die erarbeiteten Lösungen werden versioniert (z.B. cvs, subversion) und am Ende des Semesters in einer Präsenzphase in Hagen vorgestellt und mit den Betreuern diskutiert.		
Inhaltliche Voraussetzung	Erfolgreiche Bearbeitung eines der Module 63512 "Sicherheit im Internet" (01866) oder 64312 "Sicherheit - Safety & Security" (01867) bzw. äquivalente Kenntnisse		
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende Studententag/e internetgestütztes Diskussionsforum		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

erfolgreich bearbeitete  
Praktikumsaufgabe

Voraussetzung

keine

## Lehrende/r

Friedrich Steimann

## Modulbeauftragte/r

Friedrich Steimann

Dauer des Moduls  
ein SemesterECTS  
10Workload  
300 StundenHäufigkeit  
alle 2 bis 3 Jahre

## Lehrveranstaltung(en)

01595 Fachpraktikum Programmiersysteme

SWS  
4

## Detaillierter Zeitaufwand

Gemeinsame Präsenzphase in Hagen und Abschlusspräsentation: 50 Stunden  
 Einarbeitung in die theoretischen Grundlagen: 25 Stunden  
 Entwicklung eines Softwaredesigns im Team: 25 Stunden  
 Planung und Durchführung der Implementierung: 150 Stunden  
 Entwurf und Implementierung von Testfällen: 50 Stunden

## Qualifikationsziele

Nach Abschluss des Moduls

- sind Studierende in der Lage, in Gruppenarbeit ein größeres Programm zu erstellen.
- können sie zu einer gegebenen Grammatik einen Scanner und Parser entwerfen.
- sind sie in der Lage, für eine gegebene Sprache Semantikprüfung und Codegenerierung zu implementieren.
- wissen sie, wozu Regressionstests wichtig sind, und können sie in Programme einbinden.

## Inhalte

Domain Specific Languages (DSLs) sind das Mittel der Wahl, um ein Problem in einem spezifischen Kontext, der Domäne, zu lösen. Sie werden so entworfen, dass sie optimal auf die Darstellung der Probleme innerhalb der Domäne zugeschnitten sind. Unterstützt wird die Entwicklung mit DSLs durch passende Entwicklungsumgebungen.

Im Rahmen des Fachpraktikums soll für eine vorgegebene DSL von Grund auf eine Entwicklungsumgebung geschaffen werden, welche das Arbeiten mit der DSL erlaubt. Die Entwicklungsumgebung soll dabei unter anderem einen Parser, einen Compiler und einen Texteditor mit Syntaxhervorhebung und Werkzeugunterstützung für das Schreiben von Code umfassen.

Zwar existieren zahlreiche Werkzeuge, welche bei der Implementierung von DSLs unterstützen, jedoch soll im Rahmen dieses Praktikums bewusst auf die Verwendung solcher Systeme verzichtet werden, um die Konzeption und Implementierung eines komplexen Programms an einem überschaubaren Beispiel zu üben.

Das Praktikum wird in Gruppen zu 4-5 Studierenden durchgeführt. Die Teams sollen sich und insbesondere ihren Entwicklungsprozess selbst organisieren. Für die Entwicklung der Software ist ein Versionskontrollsystem einzusetzen, welches vom Lehrgebiet zur Verfügung gestellt wird. Die Ergebnisse werden in einem Abschlussgespräch per Webkonferenz präsentiert.

## Ergänzende Literatur:

Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Monica S. Lam. Compilers: Principles, Techniques, and Tools, Second Edition (Pearson Education, Inc, 2006)

Inhaltliche  
Voraussetzung

Das Fachpraktikum richtet sich an Studierende mit Interesse an anwendungsorientierter, objektorientierter Programmierung. Die Aufgabe ist dergestalt aufgebaut, dass unterschiedlichste Problemfelder berührt werden, wie etwa Parser, Semantikprüfer, Codegeneratoren, Tests – Interesse in Richtung eines dieser Felder wird also benötigt.

Weiterhin wird der sichere Umgang mit wenigstens einer bekannten, objektorientierten Programmiersprache, wie zum Beispiel Java (der über eine Belegung des Kurses 01618 hinausgeht und beispielsweise in einem Programmierpraktikum erworben wurde), vorausgesetzt.

Lehr- und Betreuungsformen	internetgestütztes Diskussionsforum Zusatzmaterial Betreuung und Beratung durch Lehrende	
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .	
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein	
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik	
<b>Prüfungsformen</b>	Art der Prüfungsleistung	Voraussetzung
Prüfung	erfolgreich bearbeitete Praktikumsaufgabe	keine
Stellenwert der Note	1/12	

Lehrende/r	Wolfram Schiffmann	Modulbeauftragte/r	Wolfram Schiffmann
Dauer des Moduls ein Semester		ECTS	10
Workload		Häufigkeit	300 Stunden in jedem Sommersemester
Lehrveranstaltung(en)	01514 Fachpraktikum Eingebettete Systeme	SS	SWS 4
Detaillierter Zeitaufwand	Bearbeiten der Heim-Versuche: 240 Stunden Vor- und Nachbereitung der Präsenzphase: 20 Stunden Präsenzphase und Studientag: 40 Stunden		
Qualifikationsziele	Die Praktikumssteilnehmerinnen und -teilnehmer sammeln praktische Erfahrungen mit einem komplexen Hardware-System und seiner Programmierung. Dadurch wiederholen und vertiefen sie den Stoff, der ihnen bereits aus grundlegenden Modulen der Technischen Informatik (z.B. 63013 "Computersysteme" und 63711 "Anwendungsorientierte Mikroprozessoren") bekannt ist. Während der Präsenzphase lernen sie, komplexe Problemstellungen zunächst in Kleingruppen zu bearbeiten und die erstellten Lösungen allen Teilnehmern zu präsentieren und zu diskutieren.		
Inhalte	<p>In diesem Fachpraktikum werden die Studierenden in die Lage versetzt, ihr in der Theorie erarbeitetes Wissen in der Praxis anzuwenden. Dafür wird ein komplexes Mikrocontroller-System inklusive der benötigten Software-Entwicklungsumgebung zur Verfügung gestellt. Jede/r Teilnehmerin/Teilnehmer wird sich im Rahmen des Praktikums intensiv mit diesem System auseinandersetzen. Die Entwicklung von Programmen für das Mikrocontroller-System erfolgt in Assemblern. Die Studierenden führen die Programmierung verschiedener Steuer- und Regelanwendungen durch, die auf den Signalen unterschiedlicher Sensoren basieren und die umfangreichen Peripherie-Komponenten des Mikrocontrollers verwenden.</p> <p><b>Ergänzende Literatur:</b>          H. Bähring: "Anwendungsorientierte Mikroprozessoren: Mikrocontroller und Digitale Signalprozessoren", Springer-Verlag, 2010</p>		
Inhaltliche Voraussetzung	Grundlagenwissen zu Mikroprozessoren, wie z.B. aus dem Modul 63013 "Computersysteme" (01608/01609) und dem Modul 63711 "Anwendungsorientierte Mikroprozessoren" (01706), sowie grundlegende Programmier-Kenntnisse.		
Lehr- und Betreuungsformen	Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Zusatzmaterial Betreuung und Beratung durch Lehrende		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .  Für die Präsenztage müssen die Teilnehmerinnen und Teilnehmer einen Laptop bzw. Rechner mitbringen, der eine Oracle-Virtual-Box-VM mit min. 2 GB Arbeitsspeicher flüssig ausführen kann.		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

erfolgreich bearbeitete  
Praktikumsaufgabe

Voraussetzung

keine

Lehrende/r	Lena Oden	Modulbeauftragte/r	Lena Oden
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01597 Fachpraktikum Parallel Programming		SWS 4
Detaillierter Zeitaufwand	Erstellung eines Pflichtenhefts für die Praktikumsaufgabe: 50 Stunden Erarbeitung eines Softwarekonzepts im Team: 50 Stunden Implementierung und Test der Software: 150 Stunden Studientage und Präsentation der Software: 50 Stunden		
Qualifikationsziele	Nachdem die Studierenden das Modul bearbeitet haben, können sie komplexe Problemstellungen in Teamarbeit lösen, parallelisierbare Komponenten identifizieren, auf die Ziel-Prozessorarchitektur verteilen, eine Softwareimplementierung für diese Rechnerarchitektur konstruieren, Testfälle generieren und damit die parallele Implementierung evaluieren, Fehler in der Implementierung identifizieren und beheben, Optimierungsmöglichkeiten gegenüberstellen und beurteilen, die Implementierung rekonstruieren und somit eine möglichst gut angepasste parallele Softwareimplementierungen für die gegebene Problemstellungen entwickeln.		
Inhalte	<p>Heutige Parallelrechner bestehen häufig aus Standard-PCs, die über ein schnelles Verbindungsnetzwerk miteinander verbunden sind. Im Fachpraktikum soll eine größere Programmieraufgabe auf einem derartigen Cluster-Computer in Gruppen von drei bis fünf Teilnehmern gelöst werden. Die Aufgabenstellung wird am Anfang des Semesters während eines Präsenztermins in Hagen bekanntgegeben und ausführlich erläutert. Außerdem wird in die Benutzung des Cluster-Computers eingeführt, es werden die Teams gebildet und Strategien zum Projektmanagement festgelegt. Bei der kooperativen Softwareentwicklung werden Versionierungssysteme verwendet. Die erarbeiteten Lösungen werden am Ende des Semesters bei einer zweiten Präsenzphase in Hagen durch eine Abschlusspräsentation vorgestellt und mit den Betreuerinnen und Betreuern diskutiert. Die Programmierung erfolgt in der Programmiersprache C/C++. Mit Hilfe der standardisierten Programmierschnittstellen PVM und MPI wird der nachrichtenbasierte Datenaustausch der parallel auf dem Cluster-Computer ablaufenden Tasks programmiert.</p> <p><b>Ergänzende Literatur:</b> Wird je nach Aufgabenstellung bekanntgegeben</p>		
Inhaltliche Voraussetzung	Erfolgreicher Abschluss des Moduls 63712 "Parallel Programming" (01727) oder der Nachweis einer gleichwertigen Qualifikation. Gute Programmierkenntnisse in C.		
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende Zusatzmaterial Studientag/e Kursmaterial		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik		



## M.Sc. Praktische Informatik

### Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

erfolgreich bearbeitete  
Praktikumsaufgabe

Voraussetzung

keine

Lehrende/r	Wolfram Schiffmann	Modulbeauftragte/r	Wolfram Schiffmann
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01515 Fachpraktikum Field Programmable Gate Arrays (FPGA)		WS SWS 4
Detaillierter Zeitaufwand	Bearbeiten der Heim-Versuche: 240 Stunden Vor- und Nachbereitung der Präsenzphase: 20 Stunden Präsenzphase und Studientag: 40 Stunden		
Qualifikationsziele	Die Praktikumsteilnehmerinnen und -teilnehmer sammeln praktische Erfahrungen mit einem komplexen Hardware-System und seiner Programmierung. Dadurch wiederholen und vertiefen sie den Stoff, der ihnen bereits aus grundlegenden Kursen der Technischen Informatik (z.B. 63013 "Computersysteme" und 63711 "Anwendungsorientierte Mikroprozessoren") bekannt ist. Während der Präsenzphase lernen sie, komplexe Problemstellungen zunächst in Kleingruppen zu bearbeiten und die erstellten Lösungen allen Teilnehmerinnen und Teilnehmer zu präsentieren und zu diskutieren.		
Inhalte	<p>Ziel dieses Praktikums ist es, die grundlegende Programmierung von FPGAs (Field Programmable Gate Arrays) mit VHDL (Very High Speed Integrated Circuits Hardware Description Language) zu erlernen. Dafür wird ein komplexes FPGA-System inklusive der benötigten Software-Entwicklungsumgebung zur Verfügung gestellt. Jede/r Teilnehmerin/Teilnehmer wird sich im Rahmen des Praktikums intensiv mit diesem System auseinandersetzen. Es werden hierbei keine Vorkenntnisse im Bereich der VHDL-Programmierung vorausgesetzt.</p> <p>Das Praktikum basiert auf dem Buch "Embedded SoPC Design with Nios II Processor and VHDL Examples" von Pong P. Chu (ISBN: 978-1-118-00888-1) 2011. Die Versuche werden mit einem Evaluations-Board DE1 der Firma Altera durchgeführt. Beides wird im Rahmen des Fachpraktikums leihweise zur Verfügung gestellt.</p>		
Inhaltliche Voraussetzung	Kenntnisse der Themen der Technischen Informatik (Schaltnetze, Schaltwerke, Boole'sche Algebra, Automaten etc.) werden allerdings als vorhanden angenommen. Sollten Sie in diesem Bereich noch Kenntnisse der Themen der Technischen Informatik (Schaltnetze, Schaltwerke, Boole'sche Algebra, Automaten etc.) werden allerdings als vorhanden angenommen. Sollten Sie in diesem Bereich noch Nachholbedarf haben, können Sie das notwendige Wissen im Modul 63013 "Computersysteme" (01608/01609) der FernUniversität Hagen erwerben.		
Lehr- und Betreuungsformen	Zusatzmaterial Betreuung und Beratung durch Lehrende Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
	Sowohl das o.g. Buch als auch das Evaluations-Board werden für den Zeitraum des Praktikums zur Verfügung gestellt. Buch und Board müssen nach Abschluss des Praktikums vollständig (d.h. inkl. Verpackung etc.), in einwandfreien Zustand zurückgegeben werden. Technische Voraussetzungen: Für die Präsenztage müssen die Teilnehmerinnen und Teilnehmer einen Laptop bzw. Rechner mitbringen, der eine Oracle-Virtual-Box-VM mit min. 2 GB Arbeitsspeicher		

flüssig ausführen kann.

**Formale Voraussetzung** mindestens zwei Wahlmodulprüfungen müssen bestanden sein

**Verwendung des Moduls** B.Sc. Informatik  
M.Sc. Informatik  
M.Sc. Praktische Informatik

<b>Prüfungsformen</b>		Art der Prüfungsleistung	Voraussetzung
Prüfung		erfolgreich bearbeitete	keine
Stellenwert der Note	1/12	Praktikumsaufgabe	

Lehrende/r	Lena Oden	Modulbeauftragte/r	Lena Oden
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01516	Fachpraktikum Scientific Programming in Python	WS SWS 4
Detaillierter Zeitaufwand	Präsenzphase am Ende des Semesters: 25 Stunden Vorbereitung und Einarbeitung: 50 Stunden Bearbeitung der vorgegebenen Programmieraufgaben: 100 Stunden Bearbeitung eines Projektes im Team: 100 Stunden Dokumentation der Ergebnisse: 25 Stunden		
Qualifikationsziele	Nach dem Abschluss des Praktikums sind die Studierenden in der Lage, mathematische, vor allem numerische Probleme, effizient mit Python zu lösen. Dies umfasst das Arbeiten mit Matrizen und Vektoren, das Lösen von Gleichungssystemen, Datenanalyse und die ansprechende Visualisierung der Ergebnisse. Sie kennen die Werkzeuge, wie diese Probleme auf moderne Hardware effizient gelöst werden können, und sind in der Lage, die Lösung auch einfach zu optimieren und zu parallelisieren. Sie kennen die wichtigsten Pakete zum effizienten arbeiten mit Python, wie numba, scipy und Pandas. Sie haben gelernt, umfangreiche Programmieraufgaben in Team zu bearbeiten.		
Inhalte	<p>Unter Scientific Computing versteht man die Umsetzung numerischer Algorithmen in eine Programmiersprache, um wissenschaftliche Probleme zu lösen. Für viele wissenschaftliche Algorithmen sind Geschwindigkeit und auch der Speicherverbrauch sehr wichtig. Reines Python hat jedoch einen hohen Speicherverbrauch und ist vergleichsweise - langsam, es bietet jedoch viele nützliche Erweiterungen an, mit deren Hilfe sich numerische Probleme effektiv lösen lassen.</p> <p>In dem Praktikum soll die effektive Nutzung dieser Pakete für wissenschaftliches Arbeiten erlernt werden. Dabei wird es zunächst schwerpunktmäßig um Numpy, ScipY, Matplotlib und Pandas gehen. Dabei wird es um die Lösung numerischer Probleme, die Analyse großer Datenmengen und das Lösen von Simulations-Aufgaben gehen. Ein weitere wichtiger Teil des Praktikums ist die ansprechende visualisierung der Ergebnisse.</p> <p>Im zweiten Teil des Praktikums geht es dann um Numba und mpi4py, welche eine effiziente Parallelisierung der Algorithmen ermöglichen.</p>		
Inhaltliche Voraussetzung	Grundlegende Programierkenntnisse Grundlegende Kenntnisse in Numerik und Analysis		
Lehr- und Betreuungsformen	Studientag/e Betreuung und Beratung durch Lehrende Zusatzmaterial		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

erfolgreich bearbeitete  
Praktikumsaufgabe

Voraussetzung

keine

Lehrende/r	Jörg Desel	Modulbeauftragte/r	Jörg Desel
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit alle 2 bis 3 Jahre
Lehrveranstaltung(en)	01593 Fachpraktikum Softwareentwicklungswerkzeuge		SWS 4
Detaillierter Zeitaufwand	Praktikum mit zwei Präsenzphasen, eine zu Beginn und eine am Ende der Veranstaltung Präsenzphasen: 50 Stunden Präsentationsvorbereitung: 10 Stunden Entwurf und Implementierung von Algorithmen im Team: 240 Stunden		
Qualifikationsziele	Erfolgreiche Absolventinnen und Absolventen verfügen über Erfahrungen in der projektorientierten Softwareentwicklung, insbesondere in den Bereichen Planung, Umsetzung und Integration verschiedener Softwaremodule in kleineren Gruppen. Da sich die Funktionalitäten der zu entwickelnden Softwaremodule an die Forschungsthemen des Lehrgebietes anlehnen, haben die Absolventinnen und Absolventen Kenntnisse im Bereich der Prozessmodellierungssprachen und sind in der Lage, auf Grundlage wissenschaftlicher Literatur effiziente Algorithmen zu implementieren.		
Inhalte	Am Lehrgebiet "Softwaretechnik und Theorie der Programmierung" werden Algorithmen und Werkzeuge zur Erstellung, Analyse, Synthese und Validierung von Prozessmodellen entworfen, implementiert und eingesetzt. Als Prozessbeschreibungssprachen werden insbesondere Petrinetze und verwandte Sprachen verwendet. Im Praktikum werden neue Algorithmen implementiert und evaluiert. Neben Erfahrungen in der Implementierung komplexer Systeme soll das Praktikum auch Kompetenz zu aktuellen Fragestellungen im angegebenen Forschungsbereich vermitteln. Erfolgreiche Absolventen und Absolventinnen haben daher eine sehr gute Grundlage für eine Abschlussarbeit in diesem Gebiet.		
Inhaltliche Voraussetzung	Vertiefte Fertigkeiten in der Programmierung mit Java. Erfolgreiche Bearbeitung des Moduls 63812 "Software Engineering" (01793). Sicherer Umgang mit mathematischer Notation. Wünschenswert sind zudem Erfahrungen mit Prozessbeschreibungssprachen wie z.B. Petrinetze, BPMN, EPK, UML activity diagrams, auch aus der beruflichen Praxis. Hilfreich ist auch der erfolgreiche Besuch des Programmierpraktikums mit einem Thema aus diesem Bereich.		
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

erfolgreich bearbeitete  
Praktikumsaufgabe

Voraussetzung

keine

Lehrende/r	André Schulz	Modulbeauftragte/r	André Schulz
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit alle drei bis vier Semester
Lehrveranstaltung(en)	01512	Fachpraktikum Theoretische Informatik	SWS 4
Detaillierter Zeitaufwand	Einarbeitung vor Beginn des Praktikums: 12 Stunden Theoretische Vorarbeiten: 10 Stunden Bearbeiten von Programmieraufgaben des ersten Abschnitts (2 Einzelabgaben): 70 Stunden Teamarbeit des zweiten Praktikumsabschnitts: 180 Stunden Erstellung einer Dokumentation: 20 Stunden Abschlusspräsentation: 8 Stunden		
Qualifikationsziele	Nach erfolgreicher Teilnahme können die Studierenden algorithmisch anspruchsvolle Aufgaben effizient lösen. Dies beinhaltet neben der Implementierungsarbeit auch die theoretische Analyse der Performanz der einzelnen Algorithmen. Des Weiteren wird den Studierenden vermittelt, wie sie die Komplexität von Problemen theoretisch sinnvoll abschätzen können (Lösbarkeit, Approximierbarkeit). Ein weiteres Ziel des Fachpraktikums ist es, Methoden aus dem Algorithm Engineering zu vermitteln.		
Inhalte	Im Fachpraktikum werden in Einzelarbeit und in Kleingruppen algorithmisch anspruchsvolle Aufgaben gelöst. In einer ersten Phase werden sich die Studierenden mit den theoretischen und technologischen Hintergründen des Problems beschäftigen. Aus den theoretischen Überlegungen werden dann praktische Algorithmen entwickelt und implementiert. Insbesondere geht es im Praktikum darum, verschiedene Strategien zum Umgang mit schweren Problemen umzusetzen.		
Inhaltliche Voraussetzung	Gute Programmierkenntnisse in Java, C++ oder Python.		
Lehr- und Betreuungsformen	internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende Einsendeaufgaben mit Korrektur und/oder Musterlösung Zusatzmaterial Lehrvideos		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik		
<b>Prüfungsformen</b>	Art der Prüfungsleistung	Voraussetzung	
Prüfung	erfolgreich bearbeitete	keine	
Stellenwert der Note	1/12	Praktikumsaufgabe	



Lehrende/r	Lars Mönch	Modulbeauftragte/r	Lars Mönch
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit alle 2 bis 3 Jahre
Lehrveranstaltung(en)	01596 Fachpraktikum Simulation von diskreten Produktionssystemen		
	SWS 4		
Detaillierter Zeitaufwand	Bearbeiten der Aufgaben des Fachpraktikums: 210 Stunden Vorbereitung und Durchführung der Präsenztage in Hagen: 90 Stunden		
Qualifikationsziele	Die Studierenden lernen, eigenständig Simulationsstudien durchzuführen. Es werden weiterhin Kenntnisse in der problemspezifischen Anpassung kommerzieller Simulationssoftware erworben. Die Studierenden sind dazu befähigt, eigenständig eine Problemanalyse durchzuführen und darauf aufbauend Simulationsmodelle zu entwickeln. Sie sind in der Lage, Simulationsmodelle eigenständig zu verifizieren und zu validieren. Die Studierenden sind dazu befähigt, die Ergebnisse der Simulationsstudie unter Verwendung einfacher statistischer Methoden auszuwerten und zu interpretieren. Die Studierenden sind in der Lage, die Ergebnisse einer Simulationsstudie überzeugend zu präsentieren. Sie erkennen die Möglichkeiten und die Grenzen der stochastischen diskreten Simulation.		
Inhalte	Im Fachpraktikum werden anhand von vorgegebenen Problemstellungen aus der Produktionsdomäne Simulationsstudien unter Verwendung eines gegebenen kommerziellen Simulators in kleinen Gruppen durchgeführt. Die zu bearbeitenden Problemstellungen sind typischerweise an praxisrelevante Fragestellungen, zumeist aus der Hochtechnologiebranche, angelehnt. Nach einer Analyse der zu lösenden Probleme werden geeignete Simulationsmodelle entwickelt. Diese werden in einem ersten Schritt verifiziert und validiert. Anschließend werden mit Hilfe dieser Simulationsmodelle die in der jeweiligen Simulationsstudie zu beantwortenden Fragen untersucht. Falls die Standardfunktionalität des Simulators nicht ausreicht, sind geeignete Erweiterungen der Funktionalität unter Verwendung der Programmiersprache C++ vorzunehmen. Die Ergebnisse sind unter Verwendung von einfachen statistischen Methoden auszuwerten und zu interpretieren. Vorschläge zur Lösung der Problemstellungen sind zu entwickeln und im Rahmen eines Vortrags vorzustellen und zu begründen.		
Inhaltliche Voraussetzung	Erfolgreiche Teilnahme am Modul 64112 „Entscheidungsmethoden in unternehmensweiten Softwaresystemen“ (01771) oder am Modul 64114 „Planungs- und Dispositionssysteme“ (01773) und insbesondere gute, auch praktische, Kenntnisse in diskreter Simulation, abgeschlossenes Grundstudium, gute Kenntnisse in objektorientierter Programmierung, Interesse an Fragestellungen aus der Produktionsdomäne		
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende Zusatzmaterial		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <a href="https://webregis.fernuni-hagen.de">https://webregis.fernuni-hagen.de</a> .		
Formale Voraussetzung	mindestens zwei Wahlmodulprüfungen müssen bestanden sein		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik		

## Prüfungsformen

Prüfung

Stellenwert  
der Note 1/12

Art der Prüfungsleistung

erfolgreich bearbeitete  
Praktikumsaufgabe

Voraussetzung

keine

Lehrende/r	Matthias Thimm	Modulbeauftragte/r	Matthias Thimm
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit regelmäßig
Lehrveranstaltung(en)	01522 Approximatives Argumentatives Schlussfolgern		regelmäßig <small>SWS</small> 4
Detaillierter Zeitaufwand	Bearbeitung des Projektes: 260 Stunden Präsentation der Ergebnisse: 40 Stunden		
Qualifikationsziele	<p>Ein Multiagentensystem (MAS) [1] ist eine Abstraktion eines Systems, in dem mehrere unabhängige Akteure (=Agenten) miteinander kooperieren und konkurrieren, um Ihre individuellen Ziele zu erreichen. Sie können beispielsweise für die Simulation realer Szenarien, wie soziale Interaktionen zwischen Personen, Strassenverkehrssituationen, oder Analyse biologischer Multiagentensystem (wie Bienen oder Ameisen) genutzt werden.</p> <p>Weitere Anwendungen finden sich beispielsweise in der Filmindustrie zur Visualisierung komplexer Szenen mit vielen Beteiligten(1) und insbesondere bei der Entwicklung von KIs in Computerspielen [2].</p> <p>Das Thema dieses Fachpraktikum ist das Erlernen theoretischer Grundlagen zu Multiagentensystemen und deren praktische Anwendung im Rahmen des aktuellen Szenarios(2) des Multi-Agent Programming Contests(3).</p> <p>In diesem Szenario konkurrieren zwei Agententeams in einer gemeinsamen Umgebung um Punkte, die bei Erfüllung gewisser Aufgaben vergeben werden. Agenten müssen dabei bestimmte Gegenstände in der Umgebung finden, aufnehmen, und ggfs. mit anderen Gegenständen kombinieren um Aufgaben erfolgreich zu erfüllen. Die Agenten eines Teams müssen dabei kooperieren um Aufgaben möglichst schnell zu erledigen und gleichzeitig konkurrieren sie mit dem anderen Team und können dieses behindern. Das Ziel des Fachpraktikums ist die Entwicklung und Implementierung von Agententeams, die im oben beschriebenen Szenario erfolgreich agieren können.</p> <p>Ergänzende Literatur:</p> <p>[1] Gerhard Weiss (Editor). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, 1999.</p> <p>[2] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, Shimon Whiteson. The StarCraft Multi-Agent Challenge. arXiv:1902.04043v5, 2019.</p>		
Inhalte	<p>Das Themengebiet des Schlussfolgerns (engl. Reasoning) ist im Bereich der Künstlichen Intelligenz einzuordnen und befasst sich mit der Ermittlung von Konklusionen aus gegebenem Wissen. Als Teilgebiet des Schlussfolgerns befasst sich die formale Argumentation mit der Interaktion zwischen Argumenten und Gegenargumenten. Obgleich verschiedene Ansätze im Bereich der formalen Argumentation existieren, liegt der Fokus dieses Fachpraktikums auf den sogenannten abstrakten Argumentationssystemen nach Dung [1]. Hierbei werden Argumentationsszenarien als gerichtete Graphen repräsentiert, wobei die Argumente als Knoten und die Konflikte zwischen ihnen als gerichtete Kanten dargestellt werden. Genauer stellt eine Kante in einem solchen Graphen einen Angriff eines Arguments auf ein anderes dar. Im Kontext solcher abstrakter Argumentationssysteme ist es üblicherweise von Interesse, sogenannte Extensionen zu identifizieren, d.h. Mengen von Argumenten, die gemeinsam akzeptierbar sind und somit eine kohärente Perspektive auf das Ergebnis der Argumentation bieten.</p> <p>Bisherige Arbeiten zur Schlussfolgerung mit Hilfe abstrakter Argumentation konzentrieren sich meist auf korrekte und vollständige Ansätze. Solche Verfahren</p>		

haben zwar den Vorteil, immer die korrekten Ergebnisse zu liefern, können aber je nach Datenlage sehr lange Berechnungszeiten mit sich bringen. Daher sollte das Potenzial von approximativen Methoden nicht außer Acht gelassen werden. Solche Verfahren haben den Vorteil, in der Regel deutlich schneller zu sein als exakte, jedoch auch den Nachteil, dass nicht garantiert werden kann, dass die Ergebnisse auch tatsächlich korrekt sind. In [3] wird eine bestimmter Typ neuronaler Netze, ein Graph Convolutional Network, trainiert, um zu entscheiden, ob Argumente in einer Extension enthalten sind oder nicht. Da die Gesamtklassifizierungsgenauigkeit nur bei etwa 80% liegt, werden in den beiden Folgearbeiten [1] und [4] anspruchsvollere Ansätze vorgestellt.

Das Ziel dieses Fachpraktikums besteht darin, zunächst die Resultate aus [1], [3], und [4] zu reproduzieren und darauf aufbauend die approximierten Resultate als Heuristik in einen exakten Ansatz, wie etwa in [5] beschrieben, einzubetten, um die Berechnungs- bzw. Lösungsgeschwindigkeit eines solchen Verfahrens zu erhöhen. Somit sollen die Vorteile beider Ansätze – Exaktheit auf der einen und hohe Geschwindigkeit auf der anderen Seite – kombiniert werden, um die praktische Anwendung von Argumentationssystemen zu erleichtern.

[1] Craandijk, D., Bex, F. (2020). Deep Learning for Abstract Argumentation Semantics. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20) (pp. 1667-1673).

[2] Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence, 77(2) (pp. 321-357).

[3] Kuhlmann, I., Thimm, M. (2019). Using Graph Convolutional Networks for Approximate Reasoning with Abstract Argumentation Frameworks: A Feasibility Study. In International Conference on Scalable Uncertainty Management (pp. 24-37). Springer, Cham.

[4] Malmqvist, L., Nightingale, P., Manandhar, S. (2020). Determining the Acceptability of Abstract Arguments with Graph Convolutional Networks. Third International Workshop on Systems and Algorithms for Formal Argumentation (SAFA2020).

[5] Geilen, L., Thimm, M. (2017). Heureka – A General Heuristic Backtracking Solver for Abstract Argumentation. In Proceedings of the 2017 International Workshop on Theory and Applications of Formal Argument (TFAFA'17).

#### Inhaltliche Voraussetzung

Grundlagen der Graphentheorie sind für das Verständnis von abstrakten Argumentationssystemen von Vorteil. Erfahrungen mit Python und Bibliotheken wie TensorFlow/PyTorch sind hilfreich und müssen ggfs. im Eigenstudium für das Praktikum gemacht werden.

#### Lehr- und Betreuungsformen

Betreuung und Beratung durch Lehrende

Kursmaterial

Einsendeaufgaben mit Korrektur und/oder Musterlösung

internetgestütztes Diskussionsforum

#### Anmerkung

Das Fachpraktikum beginnt mit einer Online-Einführungsveranstaltung, bei der eine Übersicht über Multiagentensysteme, sowie die zu bearbeitende Problemstellung gegeben wird. Die Praktikumsmitglieder werden in kleineren Teams (jeweils 4-5 Studierende) aufgeteilt, die jeweils unabhängig voneinander ein Agententeam für das Szenario entwickeln und implementieren. In regelmäßigen Abständen (ca. monatlich) wird ein Turnier abgehalten, bei dem die verschiedenen Agententeams gegeneinander antreten. Die dabei gemachten Erfahrungen sollen in die Weiterentwicklung der einzelnen Agententeams einfließen. Am Ende des Praktikums ist die entwickelte Software und eine schriftliche Ausarbeitung abzugeben. In einer finalen Veranstaltung hält jeder Teilnehmer einen Vortrag von ca. 10 Minuten zu einem Teilaspekt des Praktikums.

#### Formale Voraussetzung

mindestens zwei Wahlmodulprüfungen müssen bestanden sein

Verwendung des Moduls B.Sc. Informatik  
M.Sc. Informatik  
M.Sc. Praktische Informatik

**Prüfungsformen**

Art der Prüfungsleistung

Voraussetzung

Prüfung

erfolgreich bearbeitete

keine

Stellenwert 1/12  
der Note

Praktikumsaufgabe

# Masterseminar und Einführung in das wissenschaftliche Arbeiten

## Masterseminar und Einführung in das wissenschaftliche Arbeiten

Lehrende/r	Lehrende der Informatik		Modulbeauftragte/r		Häufigkeit in jedem Semester	
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden	Simone Opel		
Lehrveranstaltung(en)	01603	Einführung in das wissenschaftliche Arbeiten (für Masterstudiengänge der Informatik)			WS/SS	SWS 2
	ca. 10	Seminare pro Semester			WS/SS	SWS 2
Detaillierter Zeitaufwand	Der Kurs wird nur in Kombination mit einem Seminar angeboten, für das Sie sich separat anmelden müssen. Statt der Lösung von Einsendearbeiten ist eine Seminararbeit anzufertigen. Die Betreuung erfolgt durch den jeweiligen Seminaranbieter.					
	Bearbeitung des Kurses "Einführung in das wissenschaftliche Arbeiten": 150 Stunden Erarbeiten der vorgegebenen Literatur und weitere Literaturrecherche: 40 Stunden Erstellen der schriftlichen Ausarbeitung: 40 Stunden Erstellen der Präsentation, Üben des Vortrags: 40 Stunden Präsenzphase: 20 Stunden					
Qualifikationsziele	Im Kurs "Einführung in das wissenschaftliche Arbeiten" vertiefen und erweitern die Studierenden ihre Grundkenntnisse zum Schreiben wissenschaftlicher Texte in der Informatik aus dem Bachelorstudium.  Der Kurs hat das Ziel, den Studierenden Kompetenzen zu vermitteln, um selbstständig ihren Forschungs- und Schreibprozess zu planen und im Anschluss daran fach- und zielgruppengerecht einen wissenschaftlichen Text der Informatik zu schreiben und zu veröffentlichen.  Gleichzeitig erfolgt die Teilnahme an einem Seminar. Hierfür ist eine gesonderte Anmeldung über WebRegis im Vorsemester erforderlich.  Nach erfolgreicher Seminarteilnahme können Studierende <ul style="list-style-type: none"> <li>- ein wissenschaftliches Thema anhand vorgegebener Literaturhinweise erarbeiten,</li> <li>- selbstständig weitere Literatur zum Thema suchen,</li> <li>- englische Informatik-Artikel lesen und verstehen,</li> <li>- Inhalte strukturieren und mit eigenen Beispielen darstellen,</li> <li>- eine schriftliche Ausarbeitung erstellen,</li> <li>- eine Bildschirmpräsentation erstellen,</li> <li>- technische Inhalte vor einem Publikum erklären,</li> <li>- auf Fragen aus dem Publikum angemessen eingehen.</li> </ul>					
Inhalte	Im Kurs „Einführung in das wissenschaftliche Arbeiten“ erweitern die Studierenden zunächst ihre Kompetenzen wissenschaftlichen Schreibens, indem sie literaturbasiert die Verzahnung von Forschungs- und Schreibprozess reflektieren und durch im Kursmaterial enthaltene Übungen auf eigene Fragestellungen anwenden. Sie beschäftigen sich vertieft mit dem Aufbau und der Struktur wissenschaftlicher Texte in der Informatik, indem sie vorhandene deutsch- und insbesondere englischsprachige Texte analysieren, überarbeiten oder eigene Textteile formulieren. Weiterhin beschäftigen sich die Studierenden mit den Unterschieden und Gemeinsamkeiten bei Aufbau und Struktur von deutschen und englischsprachigen wissenschaftlichen Texten, zudem erweitern sie ihre Kenntnisse über Forschungsdatenmanagement und Veröffentlichungsstrategien.					

Seminare sind einem bestimmten Thema gewidmet und haben eine begrenzte

Teilnehmerzahl. Studierende bekommen von den Betreuenden ein Thema und Literaturhinweise. Sie lesen die angegebene Literatur und suchen selbstständig weitere Literatur zum Thema. Bei Rückfragen zum Seminartext wenden Sie sich bitte an die Betreuerin oder den Betreuer Ihres Seminars. Sie erstellen eine schriftliche Ausarbeitung. Oft ist vorher eine Gliederung mit der Betreuerin oder dem Betreuer abzusprechen. Sie erstellen eine Präsentation, die während der Präsenzphase vorgestellt wird. Die Betreuenden geben auch dafür Hinweise.

Inhaltliche  
Voraussetzung

keine

Lehr- und  
Betreuungsformen

Betreuung und Beratung durch Lehrende  
Kursmaterial

Anmerkung

Literaturempfehlung: Hilary Glasman-Deal: "Science Research Writing - for Non-Native Speakers of English", Imperial College Press, 2010

Für die Teilnahme an einem Seminar ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich:  
<https://webregis.fernuni-hagen.de>.

Formale Voraussetzung

mindestens vier Wahlmodulprüfungen müssen bestanden sein

Verwendung des Moduls

M.Sc. Informatik  
M.Sc. Praktische Informatik

### Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

erfolgreiche Seminarteilnahme  
(Ausarbeitung und Vortrag)

erfolgreiche Bearbeitung des Kurses  
"Einführung in das wissenschaftliche Arbeiten  
(für Masterstudiengänge der Informatik)"

Stellenwert  
der Note 1/12



# Abschlussmodul

# Abschlussmodul

Lehrende/r	Lehrende der Fakultät MI	Modulbeauftragte/r	Lehrende der Fakultät MI	
	Dauer des Moduls ein Semester	ECTS 30	Workload 900 Stunden	Häufigkeit ständig
Lehrveranstaltung(en)				
Detaillierter Zeitaufwand	Die Verteilung des Aufwands kann je nach Thema, Vorkenntnissen, Art der Arbeit und Stil der Betreuung stark variieren, hier nur ein Beispiel: <ul style="list-style-type: none"><li>• Einarbeitung in die Thematik der Abschlussarbeit, Literaturrecherche und -studium, Vertrautmachen mit existierender Softwareumgebung: 160 Stunden</li><li>• Gesprächstermine mit der Betreuerin / dem Betreuer und Mitarbeitern des betreuenden Lehrgebiets (mit Anreise), Abschlusspräsentation (u.a. abhängig von Entfernung, Art der Kommunikation): 4-8 mal 4-8 Stunden, also etwa 20 - 70 Stunden</li><li>• Eigene Entwicklung, Analyse des Problems, Entwurf, Implementierung, Tests und Experimente: 500 Stunden</li><li>• Erstellen der schriftlichen Ausarbeitung und der Abschlusspräsentation: 200 Stunden</li></ul>			
Qualifikationsziele	Die Abschlussarbeit zeigt, dass die Kandidatin oder der Kandidat gründliche Fachkenntnisse erworben hat und in der Lage ist, innerhalb einer vorgegebenen Frist ein Problem aus dem Fach selbständig mit wissenschaftlichen Methoden zu bearbeiten. Dazu gehört das Erfassen und Auswerten der relevanten Literatur. In Einzelfällen kann das Thema einer solchen Arbeit darin bestehen, die existierende Literatur zu sichten und strukturiert darzustellen. Normalerweise wird aber auch eine eigenständige kreative Leistung dazugehören, also z.B. das Entwickeln neuer Algorithmen, die (Weiter-) Entwicklung theoretischer Modelle, die Entwicklung und prototypische Implementierung neuartiger Software-Komponenten. Die Masterabsolventin oder der Masterabsolvent hat mit bestandener Abschlussarbeit demonstriert, dass sie/er in der Lage ist, die erlernten wissenschaftlichen Kenntnisse und Methoden der Informatik selbständig auf neue Problemstellungen anzuwenden.			
Inhalte	<p>Das Abschlussmodul besteht aus einer Abschlussarbeit und einem Kolloquiumsvortrag vor dem Betreuer oder der Betreuerin und weiteren Zuhörern. Im Kolloquium werden die Inhalte und Ergebnisse der Abschlussarbeit dargestellt und gegen mögliche Einwände verteidigt.</p> <p>Die Lehrenden bieten Themen an, entweder von sich aus (auch z.B. über die Webseiten des Lehrgebiets) oder auf Anfrage von Studierenden. In einer Vorbereitungsphase finden relativ viele Diskussionen zwischen Betreuenden und Studierenden statt, während die Studierenden sich in das Thema einarbeiten. In dieser Phase sind normalerweise Treffen in Hagen erforderlich, zu denen die Studierende also anreisen müssen. Auch später sind gewöhnlich direkte Gespräche zwischen den Studierenden und Betreuenden in Hagen nötig. Daneben erfolgt die Kommunikation zwischen den Studierenden und der Betreuerin oder dem Betreuer natürlich auch über E-Mail oder Telefon, teilweise auch über computergestützte Videokonferenztechniken.</p> <p>Die technische Umgebung für die Durchführung von Abschlussarbeiten wird von der FernUniversität bei Bedarf bereitgestellt. So gibt es einen Pool von Laptops, die für Abschlussarbeiten an Studierende verliehen werden. Das betreuende Lehrgebiet stellt die für die Durchführung der Aufgabe benötigte spezielle Software-Umgebung bereit. Die Studierenden erarbeiten die Literatur, entwickeln eigene kreative Beiträge (Algorithmen, Modelle, Beweise, Software-Prototypen). Sie/er beschreibt den Literatur-Hintergrund und ihre/seine eigenen Beiträge in einer schriftlichen Ausarbeitung. Die Arbeit wird nach Abgabe vom Betreuenden und einer weiteren Professorin oder einem weiteren Professor begutachtet und bewertet.</p>			

Die abschließende Präsentation im Kolloquium wird ähnlich wie ein Seminarvortrag erarbeitet, wobei die Inhalte und das Umfeld der Arbeit dargestellt werden. Eine gesonderte schriftliche Ausarbeitung ist nicht mehr erforderlich, da ja die Abschlussarbeit vorliegt. Die Präsentation wird analog zu einem Seminar, gewöhnlich per Laptop und Projektor durchgeführt. Dabei kann auch vom Studierenden erstellte Software vorgeführt werden.

Inhaltliche Voraussetzung	Inhalte und Fähigkeiten der vorausgehenden Master- und Bachelorstudien	
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende	
Anmerkung	-	
Formale Voraussetzung	alle Leistungen mit Ausnahme von höchstens zwei Modulabschlussprüfungen müssen bestanden sein	
Verwendung des Moduls	M.Sc. Informatik M.Sc. Praktische Informatik	
<b>Prüfungsformen</b>	<b>Art der Prüfungsleistung</b>	<b>Voraussetzung</b>
Prüfung	bestandene Abschlussarbeit mit Kolloquium	keine
Stellenwert der Note	1/4	

# Inhaltsverzeichnis

<b>Katalog B</b>	3
Lineare Optimierung	4
Übersetzerbau	6
Data Mining	8
Architektur und Implementierung von Datenbanksystemen	10
Verteilte Systeme	12
Einführung in Mensch-Computer-Interaktion	14
Interaktive Systeme	16
Simulation	18
Anwendungsorientierte Mikroprozessoren	19
Parallel Programming	21
Betriebliche Informationssysteme	23
Entscheidungsmethoden in unternehmensweiten Softwaresystemen	25
Wissensbasierte Systeme	27
Mobile Security	29
<b>Katalog M1: Formale Methoden und Algorithmik</b>	31
Mathematische Grundlagen der Kryptografie	32
Effiziente Graphenalgorithmien	34
Algorithmische Geometrie	36
Komplexitätstheorie	38
Effiziente Algorithmen	40
<b>Katalog M2: Computersysteme</b>	41
Betriebssysteme	42
Information Hiding	44
Virtuelle Maschinen	46
Advanced Parallel Computing	48
PC-Technologie	50
Kommunikations- und Rechnernetze	52
Sicherheit - Safety & Security	54
<b>Katalog M3: Informationssysteme</b>	56
Computerunterstütztes kooperatives Arbeiten und Lernen	57

Gestaltung Kooperativer Systeme	59
Informationsvisualisierung im Internet	61
Dokumenten- und Wissensmanagement im Internet	63
Multimedialinformationssysteme	66
Information Retrieval	68
Intelligente Informationssysteme für industrielle Anwendungen	70
Deduktions- und Inferenzsysteme	72
Methoden der Wissensrepräsentation und -verarbeitung	74
<b>Katalog M4: Softwaretechnik und Programmierung</b>	76
Objektorientierte Programmierung	77
Moderne Programmier Techniken und -methoden	79
<b>Fachpraktika</b>	81
Fachpraktikum CSCW	82
Fachpraktikum Mensch-Computer-Interaktion	84
Fachpraktikum für sichere kollaborative Anwendungen	86
Fachpraktikum Multimedia- und Internetanwendungen	88
Fachpraktikum IT-Sicherheit	90
Fachpraktikum Programmiersysteme	92
Fachpraktikum Eingebettete Systeme	94
Fachpraktikum Parallel Programming	96
Fachpraktikum Field Programmable Gate Arrays	98
Fachpraktikum Scientific Programming in Python	100
Fachpraktikum Softwareentwicklungswerkzeuge	102
Fachpraktikum Theoretische Informatik	104
Fachpraktikum Simulation von diskreten Produktionssystemen	105
Fachpraktikum Künstliche Intelligenz	107
<b>Masterseminar und Einführung in das wissenschaftliche Arbeiten</b>	110
<b>Abschlussmodul</b>	113