

Modulhandbuch

B.Sc. Informatik

FernUniversität in Hagen
Fakultät für Mathematik und Informatik

Stand:
19.05.2021

Inhaltsverzeichnis

Pflichtmodule (Studieneingangsphase)	3
Pflichtmodule	16
Katalog B	26
Katalog N	53
Fachpraktika	68
Bachelorseminar	95
Abschlussmodul	98
<i>Detalliertes Inhaltsverzeichnis</i>	101

Pflichtmodule (Studieneingangsphase)

Lehrende/r	Silke Hartlieb Michael-Ralf Skrzypek	Modulbeauftragte/r	Michael-Ralf Skrzypek
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01141 Mathematische Grundlagen		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 Klausurvorbereitung (Studientag und Selbststudium): 55 Stunden		
Qualifikationsziele	Die Studierenden entwickeln Vertrautheit mit grundlegenden Konzepten der Mathematik, sehen den zum Teil aus der Schule bekannten Stoff in neuen Zusammenhängen und lernen die Grundbegriffe und -techniken sicher zu beherrschen. Sie erlernen mathematische Arbeitsweisen, entwickeln mathematische Intuition und üben deren Umsetzung in präzise Begriffe ein. Ferner erwerben sie Basiswissen und Fertigkeiten für das gesamte weitere Studium. Durch die Teilnahme an Internet-Diskussionsgruppen sowie an den optionalen Präsenzveranstaltungen wird Teamarbeit und das Einüben wissenschaftlicher Kommunikation gefördert.		
Inhalte	<p>Das Modul besteht aus einem Kurs mit sieben Kurseinheiten und bietet eine Einführung in die mathematische Argumentation sowie einen Einblick in zentrale Themen der Linearen Algebra, Analysis und Logik.</p> <p>Nach einer Einführung in wissenschaftliche Arbeitstechniken, elementare Aussagenlogik und Beweisprinzipien werden in den ersten drei Kurseinheiten Themen der Linearen Algebra behandelt. Zu nennen sind Matrizenrechnung, elementare Zeilenumformungen von Matrizen, Existenz und Eindeutigkeit der Treppennormalform einer Matrix, Lösungsverfahren für lineare Gleichungssysteme, endlich erzeugte Vektorräume und lineare Abbildungen sowie der Zusammenhang zwischen abstrakten endlich erzeugten Vektorräumen und ihren Koordinatenräumen, beziehungsweise linearen Abbildungen und ihren Matrixdarstellungen.</p> <p>Die folgenden drei Kurseinheiten widmen sich den Grundlagen der Analysis. Hier sind zu nennen reelle Zahlen, Folgen, Reihen, Stetigkeit, Differenzierbarkeit, Taylorentwicklung, Potenzreihen und das Riemann Integral.</p> <p>In der letzten Kurseinheit wird in die Grundlagen der Aussagen- und Prädikatenlogik eingeführt.</p>		
Inhaltliche Voraussetzung	keine		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum fachmentorielle Betreuung (Regional- und Studienzentren) Studientag/e Zusatzmaterial Betreuung und Beratung durch Lehrende		
Anmerkung	-		

Formale Voraussetzung keine

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

Prüfungsformen	Art der Prüfungsleistung	Voraussetzung
Prüfung	bestandene unbenotete	keine
Stellenwert der Note	- Prüfungsklausur	

Lehrende/r	Winfried Hochstättler	Modulbeauftragte/r	Winfried Hochstättler
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01142	Algorithmische Mathematik	WS/SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (7 mal 18 Stunden): 126 Stunden Einüben des Stoffes, insbes. Durch Einsendeaufgaben (7 mal 15 Stunden): 105 Stunden Wiederholung und Klausurvorbereitung (Studientag und Selbststudium): 69 Stunden		
Qualifikationsziele	Nach erfolgreichem Abschluss des Moduls können die Studierenden abstrakte Zusammenhänge formal analysieren und logisch und formal korrekt schließen. Sie sind in der Lage, algorithmische Probleme zu modellieren und zu lösen. Die Studierenden besitzen Grundkenntnisse der Graphentheorie, der Algorithmen auf Graphen, deren Datenstrukturen und der Analyse der Laufzeit und von Optimierungsalgorithmen. Die Studierenden haben elementare Kenntnisse numerischer Berechnungen. Sie sind sensibilisiert, bei numerischen Ergebnissen rundungsfehlerbehaftete Berechnungen zu berücksichtigen.		
Inhalte	Das Modul "Algorithmische Mathematik" setzt sich zusammen aus zwei Teilen, der "Diskreten Mathematik" und der "Numerik und Optimierung". Im Kurs werden zunächst Beweismethoden an einfachen Beispielen vorgestellt und anhand von kombinatorischen Problemen eingeübt. Dabei werden elementare Abzählprobleme und Abschätzungen für Fakultäten und Binomialkoeffizienten vorgestellt. Dann werden Graphen eingeführt und als algorithmische Probleme Breitensuche, Eulertouren, Erkennen von Valenzsequenzen, minimale aufspannende Bäume und bipartites Matching diskutiert. Im numerischen Teil stellen wir die Kodierung von Zahlen vor, mögliche Fehlerquellen bei rundungsfehlerbehafteten Rechnungen und klassische Verfahren der Linearen Algebra wie LU-Zerlegung und Cholesky-Faktorisierung. In der nicht-linearen Optimierung stellen wir notwendige und hinreichende Bedingungen für Extremwerte im unrestringierten und im restringierten Fall vor. Wir diskutieren Suchverfahren, das Newton-Verfahren und das konjugierte-Gradienten-Verfahren. Aus der linearen Optimierung stellen wir den Simplex-Algorithmus vor.		
	Ergänzende Literatur: Jiri Matousek und Jaroslav Nesetril: Diskrete Mathematik - Eine Entdeckungsreise. Springer-Verlag, 2. Aufl., 2007 R.L. Graham, D. E. Knuth und O. Patashnik: Concrete Mathematics - A Foundation for Computer Science. Addison-Wesley, 2nd Edition, 1994 R. Schaback und H. Wendland: Numerische Mathematik. Springer-Verlag, 5. Aufl., 2005 W. Struckmann und D. Wätjen: Mathematik für Informatiker. Spektrum Akademischer Verlag, 2006 D. G. Luenberger: Linear and Nonlinear Programming. Addison-Wesley, 3rd Edition, 2010		
Inhaltliche Voraussetzung	Der Kurs setzt die Inhalte des Moduls 61111 "Mathematische Grundlagen" (01141) bzw. 31101 "Grundlagen der Wirtschaftsmathematik und Statistik" (40600/46001) voraus.		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum		

Anmerkung Studientag/e
Zusatzmaterial
fachmentorielle Betreuung (Regional- und Studienzentren)
-
Formale Voraussetzung keine

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

Prüfungsformen Art der Prüfungsleistung Voraussetzung
Prüfung bestandene benotete Prüfungsklausur keine
Stellenwert 1/16
der Note

Lehrende/r	Jörg Keller Lena Oden	Modulbeauftragte/r	Jörg Keller Lena Oden	
	Dauer des Moduls ein oder zwei Semester	ECTS 10	Workload 300 Stunden	Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01608 Computersysteme I			WS/SS 1.Hälfte SWS 2+1
	01609 Computersysteme II			WS/SS 2.Hälfte SWS 2+1
Detaillierter Zeitaufwand	Kurseinheiten: 150 Stunden Einsendearbeiten: 75 Stunden Studientage u. Prüfungsvorbereitung: 75 Stunden			
Qualifikationsziele	<p>Nach erfolgreicher Teilnahme können die Teilnehmerinnen und Teilnehmer Schaltfunktionen algebraisch beschreiben und Wertetabellen sowie Primimplikanten bestimmen. Sie können Schaltnetze wichtiger Schaltfunktionen wie Addierer oder Multiplexer aus parametrisierten Definitionen ableiten und ihre Komplexität analysieren. Sie können Automaten und komplexe Schaltwerke synthetisieren und analysieren.</p> <p>Nach erfolgreicher Teilnahme können die Teilnehmerinnen und Teilnehmer die Teile eines Computersystems identifizieren und ihre Funktionsweise erklären. Sie können Architekturkonzepte erläutern, insbesondere Zusammenhänge zwischen Befehlssatz- und Mikroarchitektur. Sie können die Hitraten verschiedener Cache-Architekturen bei gegebener Zugriffssequenz bestimmen. Sie können verschiedene Formen der Parallelität zur Leistungssteigerung darstellen.</p>			
Inhalte	<p>Der Kurs 01608 führt in die Grundlagen ein, die zum Verständnis von Mikroprozessoren und Rechnersystemen notwendig sind. Dies sind Schaltfunktionen und Schaltnetze, Speicherglieder und Schaltwerke, komplexe Schaltwerke. Schaltfunktionen werden algebraisch durch Boole'sche Ausdrücke beschrieben, hierbei werden auch kürzeste Normalformen bestimmt. Weiterhin wird die Realisierung von Schaltfunktionen durch Gatter sowie viel benutzte Schaltnetze vorgestellt. Als Kombination von Schaltnetzen mit Speichergliedern entstehen Schaltwerke, die als endliche Automaten beschrieben, analysiert und synthetisiert werden. Für komplexe Schaltwerke mit einer sehr großen Zahl von Zuständen erfolgt eine Beschreibung mittels ASM-Diagrammen und eine Aufteilung der Realisierung Operations- und Steuerwerk.</p> <p>Der Kurs 01609 beschreibt den Aufbau eines Computers und das Zusammenspiel seiner Komponenten, die einem komplexen Schaltwerk entsprechen. Dabei werden verschiedene Ausführungsvarianten zur Realisierung der (Teil)Schaltwerke erläutert. Dann wird gezeigt, welchen Einfluss die Befehlssatzarchitektur eines Prozessors auf dessen Hardware-Implementierung hat und es wird anhand des MIPS-Prozessors ein typisches Beispiel für den Befehlssatz eines so genannten Reduced Instruction Set Computer (RISC) vorgestellt. Danach werden verschiedene Mikroarchitekturen für Prozessoren untersucht und Beispiele von drei Mikroarchitektur-Typen für exemplarische Befehle des zuvor vorgestellten MIPS-Befehlssatzes werden ausführlich behandelt. Zum Schluss werden Möglichkeiten zur Optimierung der Speicherarchitektur vorgestellt und am Beispiel der drei möglichen Cache-Varianten analysiert.</p> <p>Ergänzende Literatur: J. Keller, W.J. Paul: Formaler Entwurf digitaler Schaltungen, 3. Auflage, Teubner-Verlag, 2005 W. Schiffmann: Technische Informatik 2 - Grundlagen der Computertechnik, 5. Auflage, Springer-Verlag, 2005</p>			

H. Bähring: Mikrorechner-Technik, Springer-Verlag, 2005
U. Brinkschulte und T. Ungerer: Mikrocontroller und Mikroprozessoren, Springer-Verlag, 2002

Inhaltliche
Voraussetzung

keine

Lehr- und
Betreuungsformen

Kursmaterial
Einsendeaufgaben mit Korrektur und/oder Musterlösung
internetgestütztes Diskussionsforum
Zusatzmaterial
Studientag/e
fachmentorielle Betreuung (Regional- und Studienzentren)
Betreuung und Beratung durch Lehrende

Anmerkung

Formale Voraussetzung keine

Verwendung des Moduls B.Sc. Informatik

Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete Prüfungsklausur keine

Stellenwert 1/16
der Note

Lehrende/r	Fabio Valdés	Modulbeauftragte/r	Jörg Desel
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01663 Datenstrukturen		WS/SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 160 Stunden Bearbeitung der Einsendeaufgaben: 80 Stunden Wiederholung und Prüfungsvorbereitung, Prüfung: 60 Stunden		
Qualifikationsziele	Nach erfolgreicher Teilnahme kennen die Studierenden die wichtigsten grundlegenden Datenstrukturen und Algorithmen der Informatik. Sie sind in der Lage, für die eigene Softwareentwicklung die jeweils geeignete Datenstruktur auszuwählen und sie ggf. anzupassen. Sie besitzen ein eingehendes Verständnis der Analyse von Algorithmen und können somit zwischen effizienten und ineffizienten Lösungen in der Programmierung unterscheiden.		
Inhalte	<p>Der Kurs behandelt grundlegende Algorithmen und Datenstrukturen der Informatik. Im Kurs werden zunächst die Begriffe Algorithmus, Datenstruktur und Datentyp erklärt und es wird die grundsätzliche Vorgehensweise bei der Analyse von Algorithmen beschrieben. Nach einer Diskussion programmiersprachlicher Basiskonzepte zur Konstruktion von Datenstrukturen werden grundlegende Datentypen (Listen, Stacks, Queues, Bäume) und ihre Implementierungen behandelt. Ein zentraler Datentyp ist das Dictionary mit seinen Implementierungen (Hashing, Suchbäume, AVL-Bäume). Weitere Datentypen zur Darstellung von Mengen sind Priority Queues und Partitionen mit MERGE und FIND Operationen. Schließlich werden Sortieralgorithmen sowie die Grundkonzepte von Graphen behandelt.</p> <p>Der zweite Teil des Kurses vermittelt Kenntnisse zu Graph-Algorithmen, geometrischen Algorithmen und Datenstrukturen, sowie zum externen Suchen und Sortieren. Zu den Graph-Algorithmen gehören etwa der Algorithmus von Dijkstra zur Bestimmung kürzester Wege, die Berechnung der transitiven Hülle eines Graphen oder eines minimalen Spannbaumes. Einen Schwerpunkt dieses Kurses bilden Algorithmen zur Behandlung geometrischer Probleme mittels Plane-Sweep und Divide-and-Conquer-Techniken. Schließlich werden B-Bäume und externe Sortierverfahren behandelt, die besonders für Datenbanksysteme von Bedeutung sind. Bei allen vorgestellten Algorithmen und Datenstrukturen steht stets die Analyse von Laufzeit und Platzbedarf im Vordergrund.</p> <p>Ergänzende Literatur</p> <p>R.H. Güting und S. Dieker, Datenstrukturen und Algorithmen. 4. Aufl., Springer-Vieweg, Wiesbaden, 2018.</p> <p>T. Ottmann und P. Widmayer, Algorithmen und Datenstrukturen. 5. Aufl., Spektrum Akademischer Verlag, Heidelberg, 2012.</p> <p>G. Saake und K.U. Sattler, Algorithmen und Datenstrukturen. Eine Einführung mit Java. 5. Aufl., dpunkt.verlag, Heidelberg, 2013.</p>		
Inhaltliche Voraussetzung	Grundkenntnisse der Programmierung sind erforderlich. Darüber hinaus sind Grundkenntnisse der Programmiersprache Java nützlich; sie können aber auch noch während der Bearbeitung des Moduls erworben werden.		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum fachmentorielle Betreuung (Regional- und Studienzentren) 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. Wirtschaftsinformatik

Prüfungsformen Art der Prüfungsleistung Voraussetzung

Prüfung bestandene benotete Prüfungsklausur keine

Stellenwert 1/16
der Note

Lehrende/r	Friedrich Steimann	Modulbeauftragte/r	Friedrich Steimann
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01618 Einführung in die objektorientierte Programmierung		WS/SS SWS 4+2
Detaillierter Zeitaufwand	Bearbeitung des Kurstextes: 150 Stunden Bearbeitung der Übungs- und Einsendeaufgaben: 100 Stunden Nachbearbeitung und Klausurvorbereitung: 50 Stunden		
Qualifikationsziele	Nach erfolgreicher Bearbeitung des Moduls - haben die Studierenden ein Grundverständnis der Konzepte der objektorientierten Programmierung. - kennen die Studierenden die wesentlichen Konstrukte der objektorientierten Programmierung am Beispiel der Programmiersprache Java. - sind die Studierenden in der Lage, unter Ausnutzung dieser Konstrukte Anwendungen in der Programmiersprache Java zu entwickeln.		
Inhalte	<p>Der Kurs gibt eine Einführung in die Konzepte der objektorientierten Programmierung und erläutert alle wichtigen Konstrukte der objektorientierten Programmiersprache Java. Zusammen mit den Einsendeaufgaben will er insbesondere auch programmiertechnisches Rüstzeug zur Eigenentwicklung von Java-Programmen vermitteln.</p> <p>Der Kurs behandelt die grundlegenden Konzepte objektorientierter Programmierung: Objekte, Klassen, Kapselung, Vererbung, Schnittstellen. Diese werden anhand ausgewählter Beispiele vorgestellt. Der Kurs zeigt auf, wie diese Konzepte zur Realisierung von Programmbausteinen und von Programmgerüsten herangezogen werden können. Zur Illustration werden wichtige Eigenschaften von Javas Klassenbibliothek zur Realisierung graphischer Bedienoberflächen erläutert. Schließlich werden ausgesuchte Aspekte zur parallelen und verteilten Programmierung mit Java behandelt.</p> <p><u>Ergänzende Literatur:</u> G. Krüger, H. Hansen, Handbuch der Java-Programmierung. Das Handbuch zu Java 8, O'Reilly, 2014 (http://www.javabuch.de/) J. Gosling, B. Joy, G. Steele, G. Bracha, A. Buckley, The Java Language Specification. Java SE 8 Edition, Addison-Wesley, 2015 https://docs.oracle.com/javase/specs/jls/se8/jls8.pdf</p>		
Inhaltliche Voraussetzung	Grundkenntnisse der imperativen Programmierung z.B. aus Modul 63811 "Einführung in die imperative Programmierung" (01613)		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Studenttag/e Zusatzmaterial fachmentorielle Betreuung (Regional- und Studienzentren) Betreuung und Beratung durch Lehrende		
Anmerkung	-		
Formale Voraussetzung	keine		

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

Prüfungsformen Art der Prüfungsleistung Voraussetzung

Prüfung bestandene benotete Prüfungsklausur keine

Stellenwert 1/16
der Note

63811

Einführung in die imperative Programmierung

Lehrende/r

Robin Bergenthum

Modulbeauftragte/r

Jörg Desel

Dauer des Moduls
ein SemesterECTS
5Workload
150 StundenHäufigkeit
in jedem Semester

Lehrveranstaltung(en)

01613 Einführung in die imperative Programmierung

WS/SS

SWS

2+1

Detaillierter Zeitaufwand

Bearbeitung der Kurseinheiten: 75 Stunden
 Lösungen der Einsendeaufgaben erstellen: 40 Stunden
 Klausurvorbereitung, Klausur: 35 Stunden

Qualifikationsziele

Nach erfolgreicher Teilnahme sind die Studierenden mit grundlegenden imperativen Programmierkonzepten vertraut. Die praktische Anwendung sämtlicher Lerninhalte beherrschen sie im Rahmen von kleineren Programmieraufgaben.

Inhalte

Kurs 01613 bildet den Einstieg in die Programmierausbildung und stellt grundlegende imperative Programmierkonzepte sowie ihre typische Anwendung vor, um kleine Programme zu entwickeln. So werden u.a. einfache und strukturierte Datentypen behandelt. Des Weiteren wird sich mit einfachen und zusammengesetzten Anweisungen und Konstrukten wie Schleifen und Funktionen befasst. Darauf aufbauend werden weitere Techniken wie z.B. Rekursion eingeführt und einfache dynamische Datenstrukturen implementiert. Zur praktischen Erläuterung und Umsetzung dieser Konzepte wird eine typisierte imperative Programmiersprache verwendet. Die in dem Kurs vermittelten imperativen Konzepte bilden auch eine wichtige Grundlage der objektorientierten Programmierung. Im Kurs wird von Beginn an Wert auf guten Programmierstil gelegt, um auf diese Weise die Erstellung von leicht lesbarem und zuverlässigem Quellcode zu fördern.

Inhaltliche
Voraussetzung

Mathematische Schulkenntnisse

Lehr- und
Betreuungsformen

Kursmaterial
 Einsendeaufgaben mit Korrektur und/oder Musterlösung
 internetgestütztes Diskussionsforum
 fachmentorielle Betreuung (Regional- und Studienzentren)

Anmerkung

-

Formale Voraussetzung

keine

Verwendung des Moduls

B.Sc. Informatik
 B.Sc. Mathematik
 B.Sc. Mathematisch-technische Softwareentwicklung

Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene unbenotete

keine

Stellenwert
der Note

-

Prüfungsklausur

63915

Einführung in die wissenschaftliche Methodik der Informatik

Lehrende/r	André Schulz	Modulbeauftragte/r	André Schulz
	Dauer des Moduls ein Semester	ECTS 5	Workload 150 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01604	Einführung in die wissenschaftliche Methodik der Informatik	WS/SS SWS 2+1
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten: 100 Stunden Bearbeiten der Einsendeaufgaben: 50 Stunden		
Qualifikationsziele	Nachdem die Studierenden das Modul bearbeitet haben, können sie die grundlegende Vorgehensweise beim wissenschaftlichen Arbeiten in der Informatik beschreiben, die Arbeitsschritte für eine konkrete Aufgabenstellung identifizieren und systematisch ausführen, die während der praktischen Arbeit erreichten Ergebnisse kategorisieren und interpretieren und in Form einer schriftlichen Ausarbeitung in LaTeX auf Bachelorniveau dokumentieren und evaluieren.		
Inhalte	In diesem Modul lernen die Studierenden die Grundzüge des wissenschaftlichen Arbeitens in der Informatik kennen. Es wird auf die geschichtliche Entwicklung der Informatik eingegangen und diskutiert, welche Teilgebiete es in der Informatik gibt, und wie diese inhaltlich abgegrenzt sind. Den Studierenden wird ein Einblick in allgemeines wissenschaftliches Arbeiten gegeben. Dies beinhaltet unter anderem Themen wie richtiges Zitieren, Empfehlungen zur Literaturrecherche und Kernpunkte der Wissenschaftsethik. Im Kurs werden außerdem Grundkenntnisse im Textsatzsystem LaTeX vermittelt.		
	Ergänzende Literatur Helmut Balzert, Marion Schröder, Christian Schäfer. Wissenschaftliches Arbeiten, 2. Auflage. W3L 2011. Nicholas Higham. Handbook of Writing for the Mathematical Sciences, 2. Auflage. SIAM 1998		
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	B.Sc. Informatik		

Prüfungsformen	Art der Prüfungsleistung	Voraussetzung
Prüfung	erfolgreich bestandene	keine
Stellenwert der Note	- Einsendeaufgaben	

Pflichtmodule

Lehrende/r	Jörg M. Haake Markus Zajac	Modulbeauftragte/r	Jörg M. Haake N. N.
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01671 Datenbanken I		WS/SS 2.Hälfte SWS 2+1
	01801 Betriebssysteme und Rechnernetze		WS SWS 2+1
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten (8 mal 20 Stunden): 160 Stunden Bearbeitung der Einsendeaufgaben inkl. Verarbeitung des Korrektur-Feedbacks (8 mal 6 Stunden): 48 Stunden Durchführung der praktischen Übungen: 10 Stunden Mitwirkung an den Diskussionen in der Kurs-Newsgroup: 22 Stunden Wiederholung und Prüfungsvorbereitung: 60 Stunden		
Qualifikationsziele	Nach erfolgreicher Teilnahme können Studierende die Grundkonzepte von Datenbanksystemen erklären. Sie sind in der Lage, ER-Diagramme für einfache Anwendungen zu erstellen und diese auf Relationenschemata abzubilden. Sie können zentrale Begriffe, wie z.B. Relation, Schlüssel, funktionale Abhängigkeit formal definieren. Sie können die Operationen der Relationenalgebra erklären und Anfragen in Algebra oder SQL formulieren. Sie können wesentliche Konzepte und Komponenten moderner Betriebssysteme benennen, erklären und vergleichen sowie ihre Abhängigkeiten analysieren und erklären. Sie können wesentliche Konzepte von Rechnernetzen und benennen und erklären sowie den Aufbau und die Funktionsweise der Schichten und Protokolle erklären und bzgl. ihrer Leistungsfähigkeit bewerten.		
Inhalte	<p>Das Modul bietet Einführungen in fundamentale Komponenten der Systemsoftware, nämlich Datenbanksysteme, Betriebssysteme und Rechnernetze. Datenbanksysteme verwalten die Daten eines Unternehmens insgesamt, unabhängig von speziellen Anwendungen. Sie bieten Benutzern eine einfache Vorstellung davon, welche Daten es gibt und wie man auf sie zugreifen kann in Form eines Datenmodells.</p> <p>Kurs 01671 behandelt die 3-Ebenen-Architektur, das relationale Modell und seine Anfragesprachen wie Relationenalgebra, Relationenkalkül und SQL. Im Bereich des Datenbankentwurfs werden das Entity-Relationship-Modell und Grundzüge der Normalisierung von Relationenschemata eingeführt.</p> <p>Kurs 01801 führt in die Grundlagen von Betriebssystemen und Rechnernetzen ein. Nach einem Überblick über den Aufgaben und Komponenten von Betriebssystemen werden Prozesse, Prozesssynchronisation und Dateisysteme als wichtige Teilbereiche betrachtet. Im zweiten Teil werden die oberen Schichten von Rechnernetzen und ihre Nutzung am Beispiel des Internets behandelt, gefolgt von Vermittlung und Sicherung in Rechnernetzen, mit einem Fokus auf den tieferen Schichten des Schichtenmodells und Aspekten des Netzwerkerns.</p> <p>Ergänzende Literatur: C.J. Date: An Introduction to Database Systems. Pearson Addison Wesley, 2004. J.D. Ullman, H. Garcia-Molina, J. Widom: Database Systems: The Complete Book. Prentice Hall, 2008. A.S. Tanenbaum, H. Bos: Moderne Betriebssysteme. 4. Auflage, Pearson Studium, 2016. A.S. Tanenbaum, D. J. Wetherall: Computernetzwerke. 3. Auflage. Pearson Studium. 2012. I.F. Kurose, K. W. Ross: Computer-Networking: A Top-Down Approach, Sixth Edition,</p>		

Inhaltliche Voraussetzung	Pearson, 2013. keine
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende Zusatzmaterial Studientag/e
Anmerkung	-
Formale Voraussetzung	mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden

Verwendung des Moduls B.Sc. Informatik

Prüfungsformen		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/16	Modulprüfung	

Lehrende/r	Lehrende der Informatik	Modulbeauftragte/r	Jörg M. Haake
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01584 Grundpraktikum Programmierung		WS/SS SWS 4
Detaillierter Zeitaufwand	Bearbeiten der Programmieraufgaben: 250 Stunden Überarbeitung der Teillösungen gemäß Rückmeldung (sofern notwendig): 40 Stunden Teilnahme an Präsenzphase: 10 Stunden		
Qualifikationsziele	Die Teilnehmenden können eine größere Programmieraufgabe selbstständig lösen und hierzu ein objektorientiertes Programm entwerfen und realisieren, welches die vorgegebene Anforderungsspezifikation erfüllt. Sie können dazu in einer modernen objektorientierten Programmierumgebung (Java, Eclipse) ein Programm implementieren, testen und dokumentieren. Die dabei anfallenden Dateien und Versionen können sie mittels eines Versionsverwaltungssystems (z.B. SVN, Git) verwalten. Sie können zudem Kommentare und Änderungsspezifikationen verstehen und umsetzen sowie fachbezogene Diskussionen mit den Betreuenden führen.		
Inhalte	Im Grundpraktikum Programmierung wird den Teilnehmenden eine Programmieraufgabe vorgegeben, anhand derer sie die Bedeutung der Modularisierung schätzen lernen. Die Teilnehmenden bearbeiten die Programmieraufgabe zu Hause. Während der Bearbeitungszeit werden ihre Fragen mit Hilfe der Praktikumsnewsgroup und per E-Mail von den Betreuenden beantwortet. Auf Basis der eingereichten Teillösungen erhalten die Teilnehmenden detailliertes Feedback und ggf. eine Aufforderung, innerhalb einer Nachfrist Änderungen durchzuführen. Am Semesterende findet eine Präsenzphase in Hagen statt, bei der die Teilnehmenden ihre Lösung vorstellen.		
Inhaltliche Voraussetzung	Für die Teilnahme am Programmierpraktikum sind fundierte Java-Kenntnisse erforderlich.		
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende Kursmaterial internetgestütztes Diskussionsforum Zusatzmaterial		
Anmerkung	Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: https://webregis.fernuni-hagen.de .		
Formale Voraussetzung	mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematisch-technische Softwareentwicklung		
Prüfungsformen	Art der Prüfungsleistung	Voraussetzung	
Prüfung	erfolgreich bearbeitete	keine	
Stellenwert der Note	1/16 Praktikumsaufgabe		

Lehrende/r	Jörg Keller	Modulbeauftragte/r	Jörg Keller
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01866	Sicherheit im Internet I	WS/SS 1.Hälfte
	01868	Sicherheit im Internet I - Ergänzungen	WS/SS 2.Hälfte
Detaillierter Zeitaufwand	Kurseinheiten: 150 Stunden Einsendearbeiten: 75 Stunden Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	Nach erfolgreicher Bearbeitung können Studierende ihre fundierten Kenntnisse von Sicherheitsmaßnahmen und -mechanismen erläutern. Sie können Wissen aus den Bereichen Mathematik, Rechnernetze und IT-Sicherheit kombinieren und mit der entstehenden Komplexität umgehen. Sie kennen die Grundlagen der wichtigsten Techniken im Bereich IT-Sicherheit und können diese darlegen, selbständig erweitern und vertiefen. Die Studierenden können unvollständig definierte Probleme aus dem Bereich IT-Sicherheit lösen und die Lösungen auch technisch umsetzen. Die Studierenden erwerben in den Übungen die Fähigkeit zu eigenverantwortlicher Tätigkeit (Einarbeitung, Analyse, Entwurf und Umsetzung).		
Inhalte	Das Modul behandelt zunächst die grundlegenden Konzepte des Themas IT-Sicherheit. Vertraulichkeit, Integrität, Authentizität und Verfügbarkeit werden als allgemeine Schutzziele vorgestellt. Konkrete Bedrohungen wie Viren und Würmer und mögliche Probleme beim Surfen im Netz werden vorgestellt. Danach werden symmetrische und asymmetrische Verschlüsselungsverfahren sowie Hash-Funktionen, digitale Unterschriften und die Grundlagen von Zertifikaten behandelt. Es wird vorgestellt, wie man sicher durch das Internet surfen kann, wie man sicher per Email kommuniziert, und wie man sicher an entfernten Rechnern arbeiten kann. Anschließend wird vorgestellt, was man zum Schutz seines privaten Rechners tun kann und soll, und worauf man bei der Konfiguration von Web-Servern achten muss. Daneben wird in diesem Teil auch auf Firewalls und Intrusion Detection Systeme, ihre Aufgaben, ihre Architektur, ihre Konfiguration und ihr Betrieb eingegangen. Den Abschluss bildet ein Abschnitt mit organisatorischen Hinweisen zum Thema Sicherheit. Schließlich werden die Themen Anonymität im Internet, aktive Inhalte, Computer-Forensik, Zugriffskontrollen, Benutzerauthentisierung und Kommunikationssicherheit in Wireless LANs und in Virtuellen Privaten Netzen behandelt.		
	Ergänzende Literatur:		
	C. Eckert: IT-Sicherheit. 9. Auflage. Oldenbourg 2014		
	R. Oppliger: Security Technologies for the World Wide Web 2nd Edition, Artech House 2003		
	B. Schneier: Applied Cryptography 2nd Edition, Wiley 1996		
Inhaltliche Voraussetzung	Modul 63012 "Softwaresysteme" (01801) oder vergleichbar		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung Betreuung und Beratung durch Lehrende internetgestütztes Diskussionsforum Studientag/e		

Anmerkung -
Formale Voraussetzung mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden

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

Prüfungsformen Art der Prüfungsleistung Voraussetzung

Prüfung bestandene benotete Prüfungsklausur keine

Stellenwert 1/16
der Note

63812

Software Engineering

Lehrende/r	Jörg Desel	Modulbeauftragte/r	Jörg Desel
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01793 Software Engineering		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 (Selbststudium): 55 Stunden		
Qualifikationsziele	Die Studierenden sind mit den vielschichtigen Problemen der Entwicklung großer (objektorientierter) Softwaresysteme vertraut. Sie kennen die systematische Ermittlung und UML-basierte Modellierung von Anforderungen sowie die schrittweise Verfeinerung bzw. Transformation der Modelle in ein ausprogrammierbares Entwurfsklassenmodell. Sie können die Lerninhalte auf der Ebene von Übungsaufgaben anwenden und mit UML-Editoren umgehen.		
Inhalte	Gegenstand des Moduls ist die methodische Entwicklung objektorientierter Softwaresysteme. Der Kurs beginnt mit einem knappen Überblick über das Gebiet Software Engineering und konzentriert sich danach auf "Informatikaspekte" der objektorientierten Softwareentwicklung. Aus Software Engineering-Sicht besteht ein wichtiger Vorteil der Objektorientierung darin, dass ihre Konzepte eine große Durchgängigkeit durch die Entwicklungsabschnitte besitzen. Dies spiegelt sich auch in der Unified Modeling Language (UML) wider, die sich als de-facto-Standard etabliert hat und deren Konstrukte in der Anforderungsermittlung und Realisierung gleichermaßen verwendet werden. Nach dem Überblick werden daher zunächst die für den Kurs relevanten UML-Elemente vorgestellt. Anschließend werden typische Arbeitsschritte der Anforderungsermittlung, der Erstellung der Softwarespezifikation und des Entwurfs besprochen, wobei die UML als Modellierungssprache benutzt wird. Der Kurs ist inspiriert von den Ideen der modellgetriebenen Softwareentwicklung, die die Entwicklung von Softwaresystemen als Folge systematischer Transformationen von Modellen begreift.		
Inhaltliche Voraussetzung	Vertiefte Kenntnisse in der objektorientierten Programmiersprache Java		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende Zusatzmaterial		
Anmerkung	-		
Formale Voraussetzung	mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematisch-technische Softwareentwicklung B.Sc. Wirtschaftsinformatik M.Sc. Wirtschaftsinformatik		

Prüfungsformen

Prüfung

Stellenwert
der Note 1/16

Art der Prüfungsleistung

bestandene benotete mündliche
Modulprüfung

Voraussetzung

keine

Lehrende/r	André Schulz	Modulbeauftragte/r	André Schulz
	Dauer des Moduls ein Semester	ECTS	Workload
		10	300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	01659 Grundlagen der Theoretischen Informatik		WS/SS SWS 4+2
Detaillierter Zeitaufwand	Das Modul besteht aus 7 Kurseinheiten. Bearbeitungszeit je Kurseinheit (inkl. Übungs- und Einsendeaufgaben): 28 Stunden (insgesamt 196 Stunden). Hinzu kommen 104 Stunden für Studientage und Prüfungsvorbereitung.		
Qualifikationsziele	Nach Bearbeiten des Kurses 01659 können die Studierenden mit den wesentlichen Grundbegriffen (Berechenbarkeit, Entscheidbarkeit, Aufzählbarkeit) umgehen. Sie können mit formalen Sprachen arbeiten und diese wichtigen Klassen zuordnen (regulär, kontextfrei, entscheidbar). Sie kennen zudem Berechnungs- und Beschreibungsmodelle dieser Sprachklassen und können mit Komplexitätsmaßen umgehen, Probleme Komplexitätsklassen zuordnen und bei schwierigen Problemen einschätzen, ob sie NP-vollständig sind. Sie lernen, wie man zeigen kann, dass Probleme nicht berechenbar sind.		
Inhalte	<p>Im ersten Kursteil wird mit Hilfe formaler Sprachen der Begriff der Berechenbarkeit entwickelt. Zunächst werden verschiedene Berechnungsmodelle vorgestellt, welche sich an der Chomsky-Hierarchie orientieren. Besonderes Augenmerk erfahren die regulären, kontextfreien und entscheidbaren Sprachen. Als Modelle werden der endliche Automat, der Kellerautomat und die Turingmaschine vorgestellt. Zudem wird auf das Konzept zur Beschreibung von Sprachen über Grammatiken vorgestellt. Dies führt zur Formulierung und Diskussion der Churchschen These.</p> <p>Der zweite Kursteil widmet sich zuerst den nichtentscheidbaren Problemen. Hier werden wichtige Probleme, wie das Halteproblem, vorgestellt und wichtige Konsequenzen (Satz von Rice, Rekursionstheorem, Postisches Korrespondenzproblem) erläutert. Auch wird auf die Entscheidbarkeit von logischen Theorien eingegangen. In diesem Zusammenhang werden auch die Gödelschen Unvollständigkeitssätze diskutiert. Anschließend wird eine Einführung in die Komplexitätstheorie gegeben. In diesem Zusammenhang werden die Komplexitätsmaße Zeit und Speicherplatz eingeführt. Mit einer eingehenden Behandlung des P-vs-NP-Problems und der NP-Vollständigkeitstheorie schließt dieser Teil.</p>		
Inhaltliche Voraussetzung	Elementare Begriffe und Methoden der Mathematik, wie sie in den einführenden Mathematikkursen des Studiengangs verwendet werden.		
Lehr- und Betreuungsformen	<p>Kursmaterial</p> <p>Einsendeaufgaben mit Korrektur und/oder Musterlösung</p> <p>internetgestütztes Diskussionsforum</p> <p>Studientag/e</p> <p>fachmentorielle Betreuung (Regional- und Studienzentren)</p> <p>Betreuung und Beratung durch Lehrende</p> <p>Zusatzmaterial</p> <p>Lehrvideos</p>		
Anmerkung	-		
Formale Voraussetzung	mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden		

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

Prüfungsformen	Art der Prüfungsleistung	Voraussetzung
Prüfung	bestandene benotete Prüfungsklausur	keine
Stellenwert der Note	1/16	

Katalog B

61412

Lineare Optimierung

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.		
	Ergänzende Literatur: 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	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematik		

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

Prüfungsformen

Prüfung

Stellenwert 1/16
der Note

Art der Prüfungsleistung

bestandene benotete mündliche
Modulprüfung

Voraussetzung

keine

Lehrende/r	Friedrich Steimann	Modulbeauftragte/r	Friedrich Steimann
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit 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	<p>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.</p> <p>Inhalte: Lexikalische Analyse (Scanner), Syntaxanalyse (Parser), Syntaxgesteuerte Übersetzung, Übersetzung einer Dokument-Beschreibungssprache, Übersetzung imperativer Programmiersprachen, Übersetzung funktionaler Programmiersprachen, Codeerzeugung und Optimierung.</p> <p>In den Übungen wird u.a. ein größeres durchgängiges Projekt zum Compilerbau bearbeitet.</p> <p>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.</p>		
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

Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden;
Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module
Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und
Softwaresysteme sind bestanden

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

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete mündliche
Modulprüfung

keine

Stellenwert
der Note

1/16

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	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
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 1/16

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 unbekanntesten 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 Studientag/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 Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden;
Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module
Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und
Softwaresysteme sind bestanden

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		Art der Prüfungsleistung	Voraussetzung
Prüfung		s. Anmerkung	keine
Stellenwert der Note	1/16		

63311

Einführung in Mensch-Computer-Interaktion

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	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		

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 1/16
der Note

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.

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	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
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 1/16
der Note

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.

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 und Wärmeleitung 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	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
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>		

Prüfungsformen

Prüfung

Stellenwert
der Note 1/16

Art der Prüfungsleistung

bestandene benotete mündliche
Modulprüfung

Voraussetzung

keine

63711

Anwendungsorientierte Mikroprozessoren

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.		
	Ergänzende Literatur: 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	-		

Formale Voraussetzung Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden;
Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module
Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und
Softwaresysteme sind bestanden

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 mündliche	keine
Stellenwert der Note	1/16	Modulprüfung	

63712

Parallel Programming

Lehrende/r

Jörg Keller
Lena Oden

Modulbeauftragte/r Lena Oden

Dauer des Moduls
ein SemesterECTS
10Workload
300 StundenHäufigkeit
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.

Ergänzende Literatur:

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, <http://ibm.com/redbooks> 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 Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden;
Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden

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		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete mündliche	keine
Stellenwert der Note	1/16	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 Architektur begriff 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	<p>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.</p> <p>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.</p>		
Inhaltliche Voraussetzung	Absolvierung des 1. und 2. Semesters ("Einführung in die WInf. 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	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		

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

Prüfungsformen

Prüfung

Stellenwert 1/16
der Note

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.

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	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden			
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 1/16
der Note

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>Ergänzende Literatur: 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 Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Studientag/e		
Anmerkung	Das Modul kann letztmalig im SS 2022 belegt werden. Eine Prüfungsteilnahme ist nur noch bis einschließlich SS 2022 möglich.		
Formale Voraussetzung	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden;		

Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden

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		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete Prüfungsklausur	keine
Stellenwert der Note	1/16		

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 Einsendeaufgaben 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	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module		

Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden

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	1/16	

Katalog N

Lehrende/r	Thomas Hering Hans-Joerg Schmerer	Modulbeauftragte/r	Thomas Hering Hans-Joerg Schmerer
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	40500 Einführung in die Betriebswirtschaftslehre		WS/SS SWS 3
	40501 Einführung in die Volkswirtschaftslehre		WS/SS SWS 3
Detaillierter Zeitaufwand	Bearbeiten der Kurseinheiten unter Nutzung des Betreuungsangebots: 180 Stunden Vorbereitung und Erbringung von Prüfungsleistungen: 120 Stunden		
Qualifikationsziele	<p>Mit dem Modul "Einführung in die Wirtschaftswissenschaft" werden im Wesentlichen die nachfolgenden Qualifikationsziele verfolgt: Die Studienanfängerinnen und Studienanfänger werden an die ökonomische Denkweise, die betriebs- und volkswirtschaftliche Fachsprache sowie die wissenschaftliche Methodik herangeführt. Eine Vielzahl elementarer betriebs- und volkswirtschaftlicher Theorien wird in einem ersten, breit angelegten Überblick kompakt vermittelt.</p>		
Inhalte	<p>Dieses Modul bietet eine Einführung in betriebs- und volkswirtschaftliche Fragestellungen.</p> <p>Einführung in die Betriebswirtschaftslehre Die Kenntnis betriebswirtschaftlicher Grundtatbestände ist eine notwendige Voraussetzung für jeden, der in Unternehmen an verantwortlicher Stelle tätig ist oder sich im Studium auf eine derartige Tätigkeit vorbereitet. Der Kurs "Einführung in die Betriebswirtschaftslehre" soll daher den Studierenden die Möglichkeit bieten, sich betriebswirtschaftliches Grundwissen anzueignen sowie betriebswirtschaftliche Methoden kennenzulernen, sie zu verstehen und anzuwenden. Dazu wird in erster Linie ein Überblick über die gesamte Breite des Faches geliefert. Nach einem einleitenden Kapitel, welches sich mit dem Gegenstand und den Zielen der Betriebswirtschaftslehre beschäftigt, wird der güterwirtschaftliche Leistungsprozess mit seinen Teildisziplinen Beschaffung, Produktion, Absatz, Organisation sowie Personal und Führung behandelt. Das dritte und abschließende Kapitel dieses Kurses befasst sich mit dem finanzwirtschaftlichen Prozess (Investition und Finanzierung, internes und externes Rechnungswesen). Wenngleich alle wesentlichen Teilbereiche der Betriebswirtschaftslehre berücksichtigt werden, erfahren einige dieser Teilbereiche eine schwerpunktmäßige Behandlung: Zur Vermittlung sowohl der wissenschaftlichen Methodik als auch der betriebswirtschaftlichen Grundlagen eignen sich besonders die Bereiche Produktion, Investition und Finanzierung sowie internes und externes Rechnungswesen.</p> <p>Einführung in die Volkswirtschaftslehre Der Kurs "Einführung in die Volkswirtschaftslehre" beschäftigt sich einleitend mit den Kernbereichen der Volkswirtschaftslehre, um den Studierenden einen Überblick über die theoretischen Fragestellungen und die Methoden der Volkswirtschaftslehre zu geben. Nach dem einleitenden ersten Teil, der die Klärung des Begriffes Volkswirtschaftslehre und die Abgrenzung zur Betriebswirtschaftslehre zum Gegenstand hat, beschäftigt sich der Kurs einleitend mit den drei Kernbereichen der Volkswirtschaftslehre, der Mikro- und Makroökonomik sowie der Wirtschaftspolitik. Der Teil II "Mikroökonomik" dieses Kurses befasst sich mit einzelwirtschaftlichen Sachverhalten, wie den individuellen Konsumententscheidungen der Haushalte und den Produktionsentscheidungen einzelner Unternehmen und deren Zusammenspiel auf Märkten. Dabei steht der</p>		

Preisbildungsprozess bei der Vielzahl der Wahlentscheidungen im Vordergrund der Analyse. Der anschließende Teil III "Makroökonomik" befasst sich hingegen mit gesamtwirtschaftlichen Aggregaten, wie z.B. dem gesamtwirtschaftlichen Güterangebot. Mit Hilfe einer modelltheoretischen Analyse werden beispielsweise folgende Fragen beantwortet: Wie entstehen Konjunkturschwankungen? Welche Rolle spielt Geld in einer Volkswirtschaft? Im abschließenden IV. Teil "Wirtschaftspolitik" wird das Handeln wirtschaftspolitischer Entscheidungsträger und deren Zielsetzung beschrieben.

Ergänzende Literatur:

Th. Hering, Ch. Toll: BWL-Klausuren, 4. Aufl., Berlin/Boston 2015.

H. Wagner, H. Turke: VWL-Klausuren. Ein Übungsbuch, 3. Auflage, De Gruyter Oldenbourg, Berlin 2017

Th. Hering, Ch. Toll: BWL kompakt, Berlin/Boston 2019.

H. Wagner, H. Turke: VWL-Klausuren. Ein Übungsbuch, 4. Auflage, De Gruyter Oldenbourg, Berlin 2020

Inhaltliche Voraussetzung

keine

Lehr- und Betreuungsformen

Kursmaterial

internetgestütztes Diskussionsforum

fachmentorielle Betreuung (Regional- und Studienzentren)

Einsendeaufgaben mit Korrektur und/oder Musterlösung

Betreuung und Beratung durch Lehrende

Zusatzmaterial

Lehrvideos

Anmerkung

-

Formale Voraussetzung

Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden;
Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden

Verwendung des Moduls B.Sc. Informatik

Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete Prüfungsklausur

s. Regelungen der Fakultät

Stellenwert der Note 1/16

Wirtschaftswissenschaft (<http://www.fernuni-hagen.de/wirtschaftswissenschaft/studium/module/31001.shtml>).

Lehrende/r	Ulrich Wackerbarth Barbara Völzmann- Stickelbrock	Modulbeauftragte/r	Ulrich Wackerbarth Barbara Völzmann- Stickelbrock
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	40560 Grundlagen und Grundbegriffe des Privatrechts		WS/SS SWS 0,6
	40561 Das Recht der Leistungsstörungen, Schadensersatz- und Vertragsrecht		WS/SS SWS 3
	40562 Recht der Kreditsicherung		WS/SS SWS 1,2
	40563 Handelsrecht		WS/SS SWS 1,2

Detaillierter Zeitaufwand Einheit / Modultitel / Workload

1. Grundlagen und Grundbegriffe des Privatrechts (30 Std.)
2. Allgemeines Schuldrecht einschließlich der Leistungsstörungen (45 Std.)
3. Kaufrecht (35 Std.)
4. Weitere vertragliche Schuldverhältnisse (35 Std.)
5. Gesetzliche Schuldverhältnisse (35 Std.)
6. Kreditsicherungsrecht (60 Std.)
7. Handelsrecht (60 Std.)

Bearbeitung der Kurseinheiten: 300 Stunden

Qualifikationsziele

Mit dem Modul werden im Wesentlichen vier Qualifikationsziele erreicht:

Die Studierenden kennen die allgemeinen Grundlagen und die Grundbegriffe des Rechts sowie wichtige Rechtsinstrumente und Rechtsinstitute in ihren rechtlichen Zusammenhängen.

Die Studierenden haben die Grundsätze und Grundprinzipien des Vertragsrechts und des Deliktsrechts erfasst und sind in der Lage, im täglichen Wirtschaftsleben auftretende Rechtsfragen sachgerecht zu beantworten.

Die Studierenden verstehen die praktisch besonders bedeutsamen Kreditsicherungsinstrumente in ihrer rechtstechnischen Konstruktion, um deren wirtschaftliche Auswirkungen zutreffend beurteilen zu können.

Die Studierenden sind im Hinblick auf die spätere Tätigkeit in einem Wirtschaftsunternehmen mit dem Sonderprivatrecht der Kaufleute vertraut.

Inhalte

Das Modul bietet eine umfassende Einführung in die im Wirtschaftsleben besonders bedeutsamen Vorschriften des Bürgerlichen Rechts und des Handelsrechts.

Gegenstand der ersten Einheit sind die im allgemeinen Teil des BGB niedergelegten Grundlagen und Grundbegriffe des Zivilrechts, deren Kenntnis Voraussetzung für das Verständnis der spezielleren Normen ist, insbesondere auch das Zustandekommen von Verträgen. Behandelt werden insbesondere die Willenserklärung, die Rechtsgeschäftslehre, die Anfechtung, die Einbeziehung allgemeiner Geschäftsbedingungen (AGB), die

Verjährung
und das Recht der Stellvertretung.

Die zweite, umfangreiche Einheit ist dem allgemeinen Schuldrecht gewidmet. Erläutert werden Begriff, Zustandekommen und Erlöschen von Schuldverhältnissen, ferner das praktisch wichtige Recht der Leistungsstörungen. Hier wird erläutert, welche Rechtsfolgen sich ergeben, wenn Verträge gar nicht, nicht fristgerecht oder nicht ordnungsgemäß durchgeführt werden.

Die dritte Einheit ist dem Kaufrecht gewidmet, die praktisch und wirtschaftlich noch immer wichtigste Form des Austauschvertrages.

Die vierte Einheit beschäftigt sich mit weiteren vertraglichen Schuldverhältnissen. Sie behandelt u.a. Miet-, Dienst- und Werkverträge. Auch moderne Vertragsformen die nicht unmittelbar gesetzlich geregelt sind, wie der Leasingvertrag, werden erläutert.

In der fünften Einheit werden die wichtigsten gesetzlichen Schuldverhältnisse dargestellt. Praktisch bedeutsam ist hier vor allem das Deliktsrecht (Schadensrecht), welches sich mit den Rechtsfolgen unerlaubter Handlungen beschäftigt.

Die sechste Einheit vermittelt zunächst Grundkenntnisse des Sachenrechts, die erforderlich sind, um das im Wirtschaftsleben wichtige Recht der Kreditsicherung verstehen zu können. Im Einzelnen werden wichtige Rechtsinstitute wie der Eigentumsvorbehalt und die Sicherungsübereignung in ihrer Konstruktion und ihrer wirtschaftlichen Bedeutung dargestellt, aber auch klassische Sicherungsmittel des Immobiliarsachenrechts wie die Hypothek und die Grundschuld.

Die siebte und letzte Einheit betrifft das im HGB geregelte Sonderprivatrecht der Kaufleute (Handelsrecht). Erläutert werden vor allem der Kaufmannsbegriff, die Firma, die Funktionsweise des Handelsregisters als auch die kaufmännischen Hilfspersonen (z. B. der Prokurist) und ihre Befugnisse. Wichtige Besonderheiten sind vor allem bei den Handelsgeschäften zu beachten. An dieser Stelle werden die Verbindungen zwischen den einzelnen Rechtsgebieten, insbesondere zum allgemeinen Teil des BGB und zum Schuldrecht besonders deutlich.

Inhaltliche
Voraussetzung

Keine speziellen Voraussetzungen

Lehr- und
Betreuungsformen

Kursmaterial
Einsendeaufgaben mit Korrektur und/oder Musterlösung
fachmentorielle Betreuung (Regional- und Studienzentren)

Anmerkung

Nicht zusammen mit dem nicht mehr angebotenen Modul "Grundlagen des Bürgerlichen Rechts" nutzbar.

Formale Voraussetzung

Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden;
Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden

Verwendung des Moduls B.Sc. Informatik

Prüfungsformen

Prüfung

Stellenwert
der Note 1/16

Art der Prüfungsleistung

bestandene benotete Prüfungsklausur

Voraussetzung

s. Regelungen der Fakultät
Wirtschaftswissenschaft (<http://www.fernuni-hagen.de/wirtschaftswissenschaft/studium/module/31061.shtml>).

Lehrende/r	Rainer Olbrich	Modulbeauftragte/r	Rainer Olbrich	
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden	Häufigkeit in jedem Semester
Lehrveranstaltung(en)	41621 Grundlagen des Marketing			WS/SS SWS 6
Detaillierter Zeitaufwand	Einheit / Titel / Workload			
	<ol style="list-style-type: none"> 1 Einführung in die Marketingplanung (25 Stunden) 2 Produktpolitik (50 Stunden) 3 Preispolitik (125 Stunden) 4 Kommunikationspolitik (25 Stunden) 5 Distributionspolitik (50 Stunden) 6 Electronic-Commerce und Online-Marketing (25 Stunden) 			
Qualifikationsziele	<p>Mit dem Modul werden im Wesentlichen die folgenden Qualifikationsziele verfolgt:</p> <ol style="list-style-type: none"> 1. Den Studierenden werden zunächst die wichtigsten konzeptionellen Grundlagen des Marketing vermittelt. Hierzu gehören insbesondere der Prozess der Marketingplanung, die Informationslieferanten und -grundlagen der Marketingplanung, die Marktsegmentierung und die Abgrenzung ‚strategischer Geschäftseinheiten‘. Im Rahmen der Marketingplanung erlernen die Studierenden z. B. die Ausgestaltung und die Vorgehensweise zur Bestimmung von Marktsegmenten, Zielgruppen und strategischen Geschäftseinheiten. Sie können exemplarische Ziele, Problembereiche und Erfolgsvoraussetzungen aufzeigen und diskutieren. 2. Die Studierenden werden befähigt, Entscheidungssituationen bezüglich des Einsatzes unterschiedlicher Marketinginstrumente nachzuvollziehen und gestalterisch zu beeinflussen. Sie können konkrete Entscheidungsprobleme formulieren, strukturieren, kritisch bewerten und anwendungsorientiert lösen. Dabei sind sie in der Lage, Interdependenzen zwischen den einzelnen Marketinginstrumenten zu erkennen und diese anhand praktischer Beispiele zu verdeutlichen. 3. Die Studierenden kennen die Nutzenkomponenten und Arten eines Produktes sowie produkt- und sortimentspolitische Basisentscheidungen und sind in der Lage, die Anwendung weiterer Gestaltungsparameter des Leistungsprogrammes, wie z. B. Markierung, Verpackung und Service, zu skizzieren. 4. Die Studierenden können eigenständig im Rahmen der statischen Preistheorie Berechnungen mit Preisabsatz-, Kosten- und Gewinnfunktionen durchführen sowie die Ergebnisse ökonomisch interpretieren. Zudem verstehen die Studierenden die Entscheidungstatbestände im Rahmen der dynamischen Preistheorie und können Spezialprobleme des Preismanagements erläutern. 5. Die Studierenden haben die Fähigkeiten, den idealtypischen Planungsprozess der Marktkommunikation aufzuzeigen sowie die zentralen Entscheidungstatbestände der Marktkommunikation darzustellen. Darüber hinaus können sie die Planung und den Einsatz der Kommunikationsinstrumente illustrieren. 6. Die Studierenden verstehen die Planungsschrittfolgen der Distributionspolitik, d.h. die wesentlichen Inhalte der Planung der Warenverkaufsprozesse und der physischen Warenverteilungsprozesse sowie deren Abwicklung und Koordination. Die Studierenden sind dadurch in der Lage, die einzelnen Planungsschrittfolgen zu erklären und zu veranschaulichen. <p>Schließlich werden die Studierenden befähigt, sich mit den Herausforderungen und Chancen des Electronic Commerce auseinanderzusetzen und Folgen der Digitalisierung für Unternehmen abzuwägen. Darüber hinaus können die Studierenden kritisch über den Einsatz ausgewählter Instrumente des Online-Marketing entscheiden.</p>			

Inhalte Dieses Modul bietet eine Einführung in die Planungsprozesse der Marketingplanung und der Marketinginstrumente. Im Vordergrund stehen dabei die wichtigsten Entscheidungsprobleme dieser Planungsbereiche.

Einheit 1: Einführung in die Marketingplanung (25 Stunden)

Im Rahmen dieser Einheit werden zunächst die konzeptionellen Grundlagen der Marketingplanung erarbeitet. Es werden dabei insbesondere der Prozess der Marketingplanung sowie die Informationslieferanten und -grundlagen der Marketingplanung erläutert. Anschließend wird die Vorgehensweise der Marktsegmentierung und der Bildung von strategischen Geschäftseinheiten dargestellt.

Einheiten 2-5: Produkt-, Preis-, Kommunikations- und Distributionspolitik (insgesamt 250 Stunden)

Im Rahmen dieser Einheiten wird die Planung der vier zentralen Instrumente des Marketing-Mix dargestellt. Hierbei handelt es sich um die Planung der Produktpolitik, der Preispolitik, der Kommunikationspolitik und der Distributionspolitik. Die Planung der Marketinginstrumente ist in die Marketingplanung eingebettet. Bei der Planung der Marketinginstrumente handelt es sich um ein eng vernetztes Planungsproblem. Daher ist eine integrierende Sichtweise zugrunde gelegt worden.

Einheit 6: Electronic Commerce und Online-Marketing (25 Stunden)

Diese Einheit widmet sich einer Einführung in den Electronic Commerce und das Online-Marketing. Hierbei stehen im ersten Themenfeld die zentralen Handlungsfelder im Vordergrund, während im zweiten Themenfeld ausgewählte Instrumente des Online-Marketing betrachtet werden.

Inhaltliche Voraussetzung Keine speziellen Voraussetzungen. Grundkenntnisse im Bereich Marketing sind hilfreich.

Lehr- und Betreuungsformen Kursmaterial
Einsendeaufgaben mit Korrektur und/oder Musterlösung
internetgestütztes Diskussionsforum
fachmentorielle Betreuung (Regional- und Studienzentren)
Betreuung und Beratung durch Lehrende

Anmerkung -

Formale Voraussetzung Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden;
Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden

Verwendung des Moduls B.Sc. Informatik

Prüfungsformen	Art der Prüfungsleistung	Voraussetzung
Prüfung	bestandene benotete Prüfungsklausur	s. Regelungen der Fakultät
Stellenwert der Note	1/16	Wirtschaftswissenschaft (http://www.fernuni-hagen.de/wirtschaftswissenschaft/studium/module/31621.shtml).

61112

Lineare Algebra

Lehrende/r	Steffen Kionke	Modulbeauftragte/r	Steffen Kionke
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Wintersemester
Lehrveranstaltung(en)	01143 Lineare Algebra		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 entwickeln Verständnis für lineare Zusammenhänge und Strukturen, erwerben vertiefte Kenntnisse im strukturellen Zugang zur Mathematik und gewinnen einen Einblick in die Anwendungen der Linearen Algebra in der Mathematik und anderen Wissenschaften. Ferner erwerben sie Basiswissen und Fertigkeiten für das gesamte weitere Studium. Durch die Teilnahme an Internet-Diskussionsgruppen sowie an den optionalen Präsenzveranstaltungen wird Teamarbeit und das Einüben wissenschaftlicher Kommunikation gefördert.		
Inhalte	Das Modul besteht aus sieben Kurseinheiten. Wesentliche Inhalte sind: <ul style="list-style-type: none"> - Äquivalenzrelationen und Faktorräume - Grundbegriffe algebraischer Strukturen: Gruppen, Ringe, Körper - symmetrische Gruppen - Polynomringe - Determinanten von Matrizen über kommutativen Ringen - charakteristisches Polynom und Minimalpolynom - Normalformproble: Diagonalisierbarkeit, nilpotente Normalform, Jordan'sche Normalform - Bilinearformen und Sesquilinearformen - Euklidische und unitäre Vektorräume, orthogonale Endomorphismen - Dualräume und adjungierte Endomorphismen 		
Inhaltliche Voraussetzung	Modul 61111 "Mathematische Grundlagen" (01141) (oder dessen Inhalt)		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Zusatzmaterial fachmentorielle Betreuung (Regional- und Studienzentren) Studientag/e Betreuung und Beratung durch Lehrende		
Anmerkung	-		
Formale Voraussetzung	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
Verwendung des Moduls	B.Sc. Informatik		

B.Sc. Mathematik

B.Sc. Mathematisch-technische Softwareentwicklung

Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

bestandene benotete Prüfungsklausur, keine

Stellenwert 1/16
der Note

2. Wh. mündl.

61211

Analysis

Lehrende/r	Delio Mugnolo	Modulbeauftragte/r	Delio Mugnolo
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01144 Analysis		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 (Studientag und Selbststudium): 55 Stunden		
Qualifikationsziele	<p>Die Studierenden entwickeln Vertrautheit mit grundlegenden Begriffen der Analysis. Insbesondere erlernen sie den Umgang mit Funktionen in höheren Dimensionen sowie die eigenständige Untersuchung der Eigenschaften einer gegebenen Funktion mehrerer Veränderlicher.</p> <p>Sie erlernen wichtige Methoden der Analysis und können mit diesen in vergleichbaren Situationen selbstständig umgehen.</p> <p>Sie erlernen vertiefte mathematische Denkweisen in konkreten und in abstrakten Situationen und sind in der Lage selbst analytische Modelle für konkrete Fragestellungen zu entwickeln und zu analysieren.</p>		
Inhalte	<p>Das Modul bietet eine Einführung in die Analysis in normierten Räumen, insbesondere im mehrdimensionalen euklidischen Raum.</p> <p>Es werden grundlegende topologische Begriffe analysiert, wie Kompaktheit, Offenheit, Abgeschlossenheit.</p> <p>Es werden Stetigkeit und Differenzierbarkeit definiert und wichtige Eigenschaften stetiger und differenzierbarer Funktionen untersucht. Wichtige Begriffe sind hierbei die partielle Ableitung, die Jacobi-Matrix und ihr Zusammenhang mit der Differenzierbarkeit.</p> <p>Der Satz von der (lokalen) Umkehrabbildung und grundlegende Begriffe der Vektoranalysis werden eingeführt. Die Grundlagen der Theorie der Kurven sowie der gewöhnlichen Differentialgleichungen werden eingeführt.</p>		
Inhaltliche Voraussetzung	Modul 61111 "Mathematische Grundlagen" (01141) oder dessen Inhalt		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Studientag/e fachmentorielle Betreuung (Regional- und Studienzentren)		
Anmerkung	-		
Formale Voraussetzung	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		

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

Prüfungsformen		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete Prüfungsklausur,	keine
Stellenwert der Note	1/16	2. Wh. mündl.	

61311

Einführung in die Stochastik

Lehrende/r

Werner Kirsch
Wolfgang Spitzer

Modulbeauftragte/r Wolfgang Spitzer

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

Lehrveranstaltung(en)

01146 Einführung in die Stochastik

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 (Studententag und Selbststudium): 55 Stunden

Qualifikationsziele

Nach Absolvierung des Moduls beherrschen die Studierenden die grundlegenden theoretischen Konzepte der Stochastik und Statistik, insbesondere in diskreten Wahrscheinlichkeitsräumen und können dies auf zielgerichtete Anwendungen übertragen. Sie sind mit verschiedenen kombinatorischen Modellen vertraut. Die Studierenden können mit Zufallsvariablen, (bedingten) Erwartungswerten und Varianzen für diskrete und absolutstetige Zufallsgrößen umgehen. Sie kennen das schwache und das starke Gesetz der großen Zahlen und verstehen die Beweise. Die Studierenden beherrschen die Poisson- und die Normalapproximation der Binomialverteilung. Mit den Grundzügen der Theorie des Schätzens und der mathematischen Tests erwerben sie einen Einblick in die mathematische Statistik und Datenanalyse.

Inhalte

Das Modul "Einführung in die Stochastik" behandelt die Themen:

- Diskreter Wahrscheinlichkeitsraum
- Axiomatik nach Kolmogorov
- Kombinatorik
- Bedingte Wahrscheinlichkeit
- stochastische Unabhängigkeit
- Zufallsvariablen
- Erwartungswerte
- höhere Momente
- Korrelationen
- Ungleichung von Tschebyschev
- schwaches und starkes Gesetz der großen Zahlen
- Satz von De Moivre und Laplace
- Einführung in die Test- und Schätztheorie

Inhaltliche
Voraussetzung

Modul 61111 "Mathematische Grundlagen" (01141) (oder dessen Inhalt)

Lehr- und
Betreuungsformen

Kursmaterial
Einsendeaufgaben mit Korrektur und/oder Musterlösung
internetgestütztes Diskussionsforum
Zusatzmaterial
Studententag/e
fachmentorielle Betreuung (Regional- und Studienzentren)
Lehrvideos

Anmerkung

-

Formale Voraussetzung

Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden;
Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module

Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden

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

Prüfungsformen		Art der Prüfungsleistung	Voraussetzung
Prüfung		bestandene benotete Prüfungsklausur,	keine
Stellenwert der Note	1/16	2. Wh. mündl.	

61511

Numerische Mathematik I

Lehrende/r	Torsten O. Linß Brice Girol Arnd Deckers	Modulbeauftragte/r	Torsten O. Linß
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01270 Numerische Mathematik I		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 (Studientag und Selbststudium): 55 Stunden		
Qualifikationsziele	<ul style="list-style-type: none"> • Fähigkeit zur mathematischen Beschreibung von Problemen, • Kenntnisse grundlegender numerischer Methoden zum exakten und näherungsweise Lösen dieser Probleme, • Bewertung der Algorithmen in Bezug auf Genauigkeit, Komplexität und Effizienz, • Fähigkeit, die zahlreichen Querverbindungen zu anderen mathematischen Gebieten zu erkennen und zu nutzen, • Basiswissen für weiterführende Veranstaltungen aus dem Bereich der angewandten Mathematik erwerben. 		
Inhalte	Fehleranalyse, lineare Gleichungssysteme und Quadratmittelprobleme, Polynome, Polynominterpolation, Quadratur, nichtlineare Gleichungen		
Inhaltliche Voraussetzung	Kenntnisse der mathematischen Grundlagen-Module		
Lehr- und Betreuungsformen	Kursmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung internetgestütztes Diskussionsforum Studientag/e Betreuung und Beratung durch Lehrende		
Anmerkung	-		
Formale Voraussetzung	Wahlmodul I: mindestens 30 von 60 ECTS der Studieneingangsphase sind bestanden; Wahlmodule II-IV: Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
Verwendung des Moduls	B.Sc. Informatik B.Sc. Mathematik B.Sc. Mathematisch-technische Softwareentwicklung		
Prüfungsformen	Art der Prüfungsleistung	Voraussetzung	
Prüfung	bestandene benotete Prüfungsklausur,	keine	
Stellenwert der Note	1/16 2. Wh. mündl.		

Fachpraktika

Lehrende/r	Mario Kubek	Modulbeauftragte/r	Mario Kubek
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Sommersemester
Lehrveranstaltung(en)	01527	Fachpraktikum Internetsicherheit	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 Internetsicherheit 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 Studierenden 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 Internetsicherheit begründen und präsentieren.		
Inhalte	In diesem Fachpraktikum werden die im Modul "Sicherheit im Internet" vermittelten Kenntnisse anhand praktischer Aufgabenstellungen angewendet werden. Jede/r Praktikumssteilnehmerin und -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 etwas größere Aufgabenstellung wie die Installation und Konfiguration eines VPN zwischen mehreren Gruppen kollaborativ gelöst. Zu Beginn des Semesters erhalten die angenommenen Studierenden die Beschreibungen der Aufgabenstellungen sowie entsprechende Literaturhinweise und Zugang zum Übungsrechner. Die erarbeiteten Lösungen werden am Ende des Semesters in einer Präsenzphase in Hagen vorgestellt und mit den Betreuern diskutiert.		
Inhaltliche Voraussetzung	- Erfolgreicher Abschluss des Moduls 63512 "Sicherheit im Internet" (Kurse 01866 und 01868) oder äquivalente Kenntnisse - Erfahrungen im Umgang mit Linux/Unix auf Shell-Ebene - Zugriff auf einen Rechner mit Internet-Zugang		
Lehr- und Betreuungsformen	internetgestütztes Diskussionsforum 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: https://webregis.fernuni-hagen.de .		
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
Verwendung des Moduls	B.Sc. Informatik		

Prüfungsformen

Prüfung
Stellenwert 1/16
der Note

Art der Prüfungsleistung

erfolgreich bearbeitete
Praktikumsaufgabe

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)	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.		
	Ergänzende Literatur: 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: https://webregis.fernuni-hagen.de .		
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		

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

Prüfungsformen	Art der Prüfungsleistung	Voraussetzung
Prüfung	erfolgreich bearbeitete	keine
Stellenwert der Note	1/16 Praktikumsaufgabe	

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: https://webregis.fernuni-hagen.de.</p>		
Formale Voraussetzung	<p>Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden</p>		
Verwendung des Moduls	B.Sc. Informatik		

M.Sc. Informatik
M.Sc. Praktische Informatik

Prüfungsformen

Prüfung
Stellenwert 1/16
der Note

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	<p>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.</p>		
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"> - Policy und Datenstrom basierte Autorisierung von Datenzugriffen. - Kollaborative Benutzungsschnittstellen. Web Basiert (Vaadin, Java, Spring, Python, Django). - Smart Contracts und Blockchain (z. B. LTO Network, Ethereum) - Internet der Dinge: Sensordaten von Maschinen (z. B. OPC-UA, MQTT) - Backend Infrastrukturen mit Microservice-Architekturen, Domain Driven Design, CQRS-ES, Java, Spring und Axon. <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: https://webregis.fernuni-hagen.de .		
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		

M.Sc. Informatik
M.Sc. Praktische Informatik

Prüfungsformen

Prüfung
Stellenwert 1/16
der Note

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: https://webregis.fernuni-hagen.de .		
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		

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

Prüfungsformen		Art der Prüfungsleistung	Voraussetzung
Prüfung		erfolgreich bearbeitete	keine
Stellenwert der Note	1/16	Praktikumsaufgabe	

Lehrende/r	Friedrich Steimann	Modulbeauftragte/r	Friedrich Steimann
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit alle 2 bis 3 Jahre
Lehrveranstaltung(en)	01595 Fachpraktikum Programmiersysteme		SWS 4
Detaillierter Zeitaufwand	<p>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</p>		
Qualifikationsziele	<p>Nach Abschluss des Moduls</p> <ul style="list-style-type: none"> - 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	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Ergänzende Literatur: Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman, Monica S. Lam. Compilers: Principles, Techniques, and Tools, Second Edition (Pearson Education, Inc, 2006)</p>		
Inhaltliche Voraussetzung	<p>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.</p> <p>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.</p>		

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: https://webregis.fernuni-hagen.de .	
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden	
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik	
Prüfungsformen	Art der Prüfungsleistung	Voraussetzung
Prüfung	erfolgreich bearbeitete Praktikumsaufgabe	keine
Stellenwert der Note	1/16	

Lehrende/r	Wolfram Schiffmann	Modulbeauftragte/r	Wolfram Schiffmann
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit 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	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.		
	Ergänzende Literatur: H. Bähring: "Anwendungsorientierte Mikroprozessoren: Mikrocontroller und Digitale Signalprozessoren", Springer-Verlag, 2010		
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: https://webregis.fernuni-hagen.de . 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	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik		

M.Sc. Praktische Informatik

Prüfungsformen

Prüfung

Stellenwert 1/16
der Note

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 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 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: https://webregis.fernuni-hagen.de .		
	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 Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden

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

Prüfungsformen		Art der Prüfungsleistung	Voraussetzung
Prüfung		erfolgreich bearbeitete	keine
Stellenwert der Note	1/16	Praktikumsaufgabe	

Lehrende/r	Jörg Keller Lena Oden	Modulbeauftragte/r	Lena Oden
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit alle 2 bis 3 Jahre
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>Ergänzende Literatur: 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: https://webregis.fernuni-hagen.de .		
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		

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

Prüfungsformen	Art der Prüfungsleistung	Voraussetzung
Prüfung	erfolgreich bearbeitete	keine
Stellenwert der Note	1/16 Praktikumsaufgabe	

Lehrende/r	Lena Oden	Modulbeauftragte/r	Lena Oden
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit regelmäßig
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: https://webregis.fernuni-hagen.de .		
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik		

M.Sc. Praktische Informatik

Prüfungsformen

Prüfung

Stellenwert
der Note 1/16

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: https://webregis.fernuni-hagen.de .		
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik		

Prüfungsformen

Prüfung
Stellenwert 1/16
der Note

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	<p>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</p>		
Qualifikationsziele	<p>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.</p>		
Inhalte	<p>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.</p>		
Inhaltliche Voraussetzung	Gute Programmierkenntnisse in Java, C++ oder Python.		
Lehr- und Betreuungsformen	<p>internetgestütztes Diskussionsforum Betreuung und Beratung durch Lehrende Einsendeaufgaben mit Korrektur und/oder Musterlösung Zusatzmaterial Lehrvideos</p>		
Anmerkung	<p>Für die Teilnahme an dieser Veranstaltung ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: https://webregis.fernuni-hagen.de.</p>		
Formale Voraussetzung	<p>Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden</p>		
Verwendung des Moduls	<p>B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik</p>		

Prüfungsformen

Prüfung
Stellenwert 1/16
der Note

Art der Prüfungsleistung

erfolgreich bearbeitete
Praktikumsaufgabe

Voraussetzung

keine

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), insbesondere auch gute Kenntnisse in diskreter Simulation und 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: https://webregis.fernuni-hagen.de .		
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden		
Verwendung des Moduls	B.Sc. Informatik M.Sc. Informatik M.Sc. Praktische Informatik		

Prüfungsformen

Prüfung
Stellenwert 1/16
der Note

Art der Prüfungsleistung

erfolgreich bearbeitete
Praktikumsaufgabe

Voraussetzung

keine

Bachelorseminar

Bachelorseminar

Lehrende/r	Lehrende der Informatik	Modulbeauftragte/r		Lehrende der Informatik
	Dauer des Moduls ein Semester	ECTS 5	Workload 150 Stunden	Häufigkeit in jedem Semester
Lehrveranstaltung(en)				
Detaillierter Zeitaufwand	Themenauswahl: 10 Stunden Erarbeiten der vorgegebenen Literatur und weitere Literaturrecherche, Lesen weiterer Artikel: 40 Stunden Erstellen der schriftlichen Ausarbeitung: 40 Stunden Erstellen der Präsentation, Üben des Vortrags: 40 Stunden Präsenzphase: 20 Stunden			
Qualifikationsziele	Nach erfolgreicher Teilnahme können Studierende <ul style="list-style-type: none">- ein wissenschaftliches Thema anhand vorgegebener Literaturhinweise erarbeiten,- selbstständig weitere Literatur zum Thema suchen,- englische Informatik-Artikel lesen und verstehen,- Inhalte strukturieren und mit eigenen Beispielen darstellen,- eine schriftliche Ausarbeitung erstellen,- eine Bildschirmpräsentation erstellen,- technische Inhalte vor einem Publikum erklären,- auf Fragen aus dem Publikum angemessen eingehen.			
Inhalte	<p>Seminare sind einem bestimmten Thema gewidmet und haben eine begrenzte Teilnehmerzahl. Wegen der Berufstätigkeit der Studierenden werden die Präsenzphasen an der FernUniversität gewöhnlich in Kompaktform an einem Wochenende in Hagen durchgeführt. Studierende bekommen von den Betreuenden ein Thema und Literaturhinweise. Sie lesen die angegebene Literatur und suchen selbstständig weitere Literatur zum Thema. Sie erstellen eine schriftliche Ausarbeitung. Oft ist vorher eine Gliederung mit der Betreuerin oder dem Betreuer abzusprechen. Sie erstellen eine Präsentation, die mit einem Laptop vorführbar ist. Die Betreuenden geben auch dafür Hinweise.</p> <p>Zur Präsenzphase des Seminars treffen sich alle Teilnehmerinnen und Teilnehmer und die Betreuenden in Hagen. Die Studierenden halten ihre Vorträge; die entsprechenden Inhalte werden - auch vortragsübergreifend - diskutiert. Die Studierenden erhalten Rückmeldung von den Betreuenden zur Qualität ihrer Präsentation und der schriftlichen Ausarbeitung.</p> <p>Seminare werden mit jeweils neuen, aktuellen Inhalten angeboten. Deshalb gibt es hier im Modulhandbuch nur eine allgemeine Beschreibung.</p>			
Inhaltliche Voraussetzung	je nach Thema			
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende			
Anmerkung	Für die Teilnahme am Seminar ist neben der Belegung ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: https://webregis.fernuni-hagen.de			
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden			

Prüfungsformen		Art der Prüfungsleistung	Voraussetzung
Prüfung		erfolgreiche Seminarteilnahme	keine
Stellenwert der Note	1/16	(Ausarbeitung und Vortrag)	

Abschlussmodul

Abschlussmodul

Lehrende/r	Lehrende der Informatik		Modulbeauftragte/r		Lehrende der Informatik	
	Dauer des Moduls ein Semester	ECTS 15	Workload 450 Stunden	Häufigkeit ständig		
Lehrveranstaltung(en)	1)	Reading Course Informatik			WS/SS	SWS 2+1
	2)	Bachelorarbeit Informatik			WS/SS	SWS 4+2
Detaillierter Zeitaufwand	Vorbereitung auf wissenschaftliches Arbeiten: 75 Stunden Literaturrecherche: 50 Stunden Erstellung eines Abschlussarbeitskonzeptes: 25 Stunden Bearbeitung des Themas: 275 Stunden Vorbereitung und Durchführung der Präsentation und des Kolloquiums: 25 Stunden					
Qualifikationsziele	Im Reading Course arbeiten sich die Studierenden in ein fortgeschrittenes Gebiet der Informatik selbstständig anhand von Büchern, Artikeln und anderer Fachliteratur ein und erstellen ein Abschlussarbeitskonzept. 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 selbstständig mit wissenschaftlichen Methoden zu bearbeiten. Die Bachelorabsolventin oder der Bachelorabsolvent hat mit bestandener Abschlussarbeit demonstriert, dass sie/er in der Lage ist, die erlernten Kenntnisse und Methoden der Informatik selbstständig auf neue Problemstellungen anzuwenden.					
Inhalte	Der Reading Course beinhaltet eine Anleitung zum selbständigen wissenschaftlichen Arbeiten und wird zur Vorbereitung in Thematiken des Umfeldes der darauffolgenden Bachelorarbeit genutzt. Die Abschlussarbeit ist eine Prüfungsarbeit in Informatik und wird komplettiert durch ihre Präsentation in einem Kolloquiumsvortrag vor den betreuenden Prüfenden. 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. 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 des vorausgehenden Bachelorstudiums					
Lehr- und Betreuungsformen	Betreuung und Beratung durch Lehrende					
Anmerkung	-					
Formale Voraussetzung	Studieneingangsphase ist abgeschlossen, die Module Grundpraktikum Programmierung, Grundlagen der Theoretischen Informatik und Softwaresysteme sind bestanden					

Verwendung des Moduls B.Sc. Informatik

Prüfungsformen

Prüfung

Stellenwert
der Note 1/8

Art der Prüfungsleistung

erfolgreiche Teilnahme am Reading
Course und bestandene
Abschlussarbeit mit Kolloquium

Voraussetzung

Vor der Vergabe eines Themas für die
Bachelorarbeit ist der Abschluss des Reading
Course durch ein positiv begutachtetes
Abschlussarbeitskonzept nachzuweisen.

Inhaltsverzeichnis

Pflichtmodule (Studieneingangsphase)	3
Mathematische Grundlagen	4
Algorithmische Mathematik	6
Computersysteme	8
Datenstrukturen und Algorithmen	10
Einführung in die objektorientierte Programmierung	12
Einführung in die imperative Programmierung	14
Einführung in die wissenschaftliche Methodik der Informatik	15
Pflichtmodule	16
Softwaresysteme	17
Grundpraktikum Programmierung	19
Sicherheit im Internet	20
Software Engineering	22
Grundlagen der Theoretischen Informatik	24
Katalog B	26
Lineare Optimierung	27
Übersetzerbau	29
Data Mining	31
Verteilte Systeme	33
Einführung in Mensch-Computer-Interaktion	35
Interaktive Systeme	37
Simulation	39
Anwendungsorientierte Mikroprozessoren	41
Parallel Programming	43
Betriebliche Informationssysteme	45
Entscheidungsmethoden in unternehmensweiten Softwaresystemen	47
Wissensbasierte Systeme	49
Mobile Security	51
Katalog N	53
Einführung in die Wirtschaftswissenschaft	54
Grundlagen des Privat- und Wirtschaftsrechts	56

Grundlagen des Marketing	59
Lineare Algebra	61
Analysis	63
Einführung in die Stochastik	65
Numerische Mathematik I	67
Fachpraktika	68
Fachpraktikum Internetsicherheit	69
Fachpraktikum CSCW	71
Fachpraktikum Mensch-Computer-Interaktion	73
Fachpraktikum für sichere kollaborative Anwendungen	75
Fachpraktikum Multimedia- und Internetanwendungen	77
Fachpraktikum Programmiersysteme	79
Fachpraktikum Eingebettete Systeme	81
Fachpraktikum Field Programmable Gate Arrays	83
Fachpraktikum Parallel Programming	85
Fachpraktikum Scientific Programming in Python	87
Fachpraktikum Softwareentwicklungswerkzeuge	89
Fachpraktikum Theoretische Informatik	91
Fachpraktikum Simulation von diskreten Produktionssystemen	93
Bachelorseminar	95
Abschlussmodul	98