

Basismodule

inf030 - Programmierung, Datenstrukturen und Algorithmen

Modulbezeichnung	Programmierung, Datenstrukturen und Algorithmen
Modulcode	inf030
Kreditpunkte	9.0 KP
Workload	270 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Basismodule • Fach-Bachelor Mathematik (Bachelor) > Nebenfachmodule • Fach-Bachelor Nachhaltigkeitsökonomik (Bachelor) > Wahlpflichtbereich • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Basiscurriculum • Fach-Bachelor Wirtschaftswissenschaften (Bachelor) > Studienrichtung Wirtschaftsinformatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Basismodule
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Sebastian Lehnhoff ◦ Dietrich Boles <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Sebastian Lehnhoff ◦ Dietrich Boles ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Ziele des Moduls: Das Programmieren ist eine der Basistätigkeiten von Informatikern und Voraussetzung für viele andere Veranstaltungen des Informatikstudiums. Ziel des Moduls „Programmierung, Datenstrukturen und Algorithmen“ ist das Erlernen grundlegender Konzepte der imperativen und prozeduralen Programmierung anhand der Programmiersprache Java sowie die Vorstellung bekannter, effizienter Algorithmen und Datenstrukturen für verschiedene, häufig vorkommende Problemstellungen. Die Studierenden sollen nach Abschluss des Moduls selbstständig imperative Programme auf der Grundlage von Java für die Lösung kleinerer Probleme entwickeln und die Effizienz ihrer Programme einschätzen können.</p> <p>Fachkompetenzen Die Studierenden: - beschreiben grundlegende Konzepte der imperativen Programmierung mit Java - erkennen die Terminologie der imperativen Programmierung und verwenden die entsprechenden Begriffe präzise bei Diskussionen - beschreiben, was ihnen vorgelegte Programme tun - entwickeln selbstständig Programme für die Lösung kleinerer Probleme - untersuchen systematisch eigene und fremde Programme auf Fehler - setzen moderne Programmientwicklungsumgebungen zum Entwickeln und Testen von Programmen ein - erstellen Algorithmen mit allgemeinen Entwurfskonzepten (z.B. Greedy-Verfahren, Divide-and-Conquer-Verfahren) - benennen Algorithmen und Datenstrukturen zur Lösung von häufig vorkommenden Problemen und bewerten diese in ihrer Anwendbarkeit - benennen Probleme der Effizienz von algorithmischen Lösungen konkreter Fragestellungen und bewerten diese - wählen fundiert einen Algorithmus und eine Datenstruktur zur Lösung eines konkreten Problems aus - wenden die gelernten Algorithmen und Datenstrukturen sinnvoll auf gegebene und konkrete Probleme an</p> <p>Methodenkompetenzen Die Studierenden - lösen gegebene Probleme unter den Gesichtspunkt der imperativen Programmierung - übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben</p> <p>Sozialkompetenzen Die Studierenden - vermitteln die Struktur und Wirkungsweise selbst entwickelter Programme an andere - präsentieren Lösungen zu kleinen Aufgaben vor Gruppen</p> <p>Selbstkompetenzen Die Studierenden - organisieren sich beim Finden von algorithmischen Lösungen für kleine und mittelgroße Probleme der Informatik - beziehen die Konzepte des allgemeinen Programmentwurfs in ihr Handeln ein.</p>
Modulinhalte	Im ersten Teil werden allgemeine Grundbegriffe der Programmierung eingeführt:

- Algorithmus, Programmiersprachen, Computer
- Entwicklungswerkzeuge, Entwicklungsphasen
- Compiler
- Syntaxdiagramme
- Logik
- Dokumentation

Der zweite Teil befasst sich mit imperativen Programmierkonzepten:

- Datentypen
- Variablen
- Ausdrücke, Anweisungen
- Kontrollstrukturen
- Prozeduren, Funktionen, Parameter
- Rekursion
- Referenzdatentypen, Arrays, Verbunde

Der dritte Teil beinhaltet eine Einführung in Datenstrukturen und Algorithmen sowie die Diskussion ihrer Effizienz, d.h. des Berechnungsaufwands in Abhängigkeit vom Umfang der zu verarbeitenden Daten. Das Modul stellt für verschiedene, häufig vorkommende Problemstellungen bekannte, effiziente Algorithmen und Datenstrukturen vor. Konkreter sind dies insbesondere:

- Verfahren zum Suchen nach Schlüssel, sowie Einfügen und Löschen in dynamischen Datenmengen, z.B. AVL-Bäume, B-Bäume, Hash-Verfahren,
- Methoden zur Suche nach Textmustern,
- Verfahren zum Sortieren von Daten nach Schlüsselwerten, z.B. Quick-Sort und Heap-Sort,
- Graph-basierte Anwendungen, z.B. zur Ermittlung kürzester Wege in Graphen,
- einfache numerische Verfahren, z.B. zum Lösen linearer Gleichungssysteme,
- ein Beispiel eines diskreten Optimierungsverfahrens, z.B. das Simplex-Verfahren.

Ergänzt wird der Vorlesungsteil um einen umfassenden Übungsteil, in dem insbesondere die vermittelten Programmierinhalte an praktischen Beispielen umgesetzt werden.

Literaturempfehlungen

Essentiell:

- Skript (wird entweder in gedruckter Form oder in elektronischer Form über das StudIP fortlaufend in der Vorlesung zur Verfügung gestellt)

Gute Sekundärliteratur:

- Dietmar Ratz, Jens Scheffler, Detlev Seese, Jan Wiesenberger: Grundkurs Programmieren in Java, Carl Hanser Verlag.
- Joachim Goll, Cornelia Heinisch: Java als erste Programmiersprache, Springer Vieweg Verlag
- Ottmann, Widmayer: Algorithmen und Datenstrukturen. Spektrum Verlag, 5. Auflage, 2012
- Sedgwick, Wayne: Algorithms. Addison Wesley, 4th ed., 2011
- Siege: Einführung in die Informatik. Shaker Verlag, 2013

Links				
Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	WiSe			
Aufnahmekapazität Modul	unbegrenzt			
Modullevel / module level	BC (Basiscurriculum / Base curriculum)			
Modulart / typ of module	Pflicht / Mandatory			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge				
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Genauere Prüfungsmodalitäten werden in der Veranstaltung bekannt gegeben.		Klausur oder mündliche Prüfung oder Portfolio	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		4.00	WiSe	56 h
Übung		2.00	WiSe	28 h
Präsenzzeit Modul insgesamt				84 h

inf031 - Objektorientierte Modellierung und Programmierung

Modulbezeichnung	Objektorientierte Modellierung und Programmierung
Modulcode	inf031
Kreditpunkte	9.0 KP
Workload	270 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Basismodule • Fach-Bachelor Mathematik (Bachelor) > Nebenfachmodule • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Basiscurriculum • Fach-Bachelor Wirtschaftswissenschaften (Bachelor) > Studienrichtung Wirtschaftsinformatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Basismodule
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Andreas Winter ◦ Dietrich Boles <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Andreas Winter ◦ Dietrich Boles ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Ziele des Moduls: Die Objektorientierung stellt heutzutage den Stand der Technik in der Softwareentwicklung dar. Gegebene Problemstellungen werden dabei mit Hilfe objektorientierter Analyse- und Entwurfsverfahren zunächst in ein objektorientiertes Modell und anschließend in ein objektorientiertes Programm überführt. Ziel des Moduls „Objektorientierte Modellierung und Programmierung“ ist das Erlernen grundlegender Konzepte der objektorientierten Modellierung mit Hilfe der UML als Modellierungsnotation und der objektorientierten Programmierung mit der Programmiersprache Java. Die Studierenden sollen nach Abschluss des Moduls selbstständig objektorientierte Programme auf der Grundlage von Java für die Lösung mittelgroßer Probleme entwickeln können.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • kennen grundlegende Konzepte der objektorientierten Modellierung und UML als Modellierungsnotation • kennen grundlegende Konzepte der objektorientierten Programmierung mit Java • kennen die Terminologie der objektorientierten Modellierung und Programmierung und verwenden die entsprechenden Begriffe präzise bei Diskussionen • können beschreiben, was ihnen vorgelegte objektorientierte Programme tun • entwickeln selbstständig Modelle und Programme für die Lösung mittelgroßer Probleme • untersuchen systematisch eigene und fremde Modelle und Programme auf Fehler • setzen moderne Entwicklungsumgebungen zum Modellieren und Entwickeln von Programmen ein • kennen die Unterschiede zwischen dem imperativen, objektorientierten, funktionalen und logischen Programmierparadigma <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • entwickeln selbstständig Programme für gegebene Probleme durch konsequente Anwendung der Konzepte der objektorientierten Modellierung und Programmierung • übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • vermitteln die Struktur und Wirkungsweise selbst entwickelter Modelle und Programme an andere • präsentieren selbstständig entwickelte Lösungen vor Gruppen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • organisieren sich beim Entwickeln von Programmen für kleine und mittelgroße Probleme der Informatik • beziehen die Konzepte des objektorientierten Programmierparadigmas in ihr Handeln ein
Modulinhalte	Im ersten Teil werden grundlegende Konzepte der objektorientierten Modellierung und Programmierung vermittelt:

- Modelle und Modellierung
- UML-Klassendiagramme
- Use-Cases
- Klassen und Objekte
- Datenkapselung
- Vererbung
- Polymorphie und dynamisches Binden
- Ausnahmebehandlung
- Generizität

Im zweiten Teil werden wichtige Konzepte und Klassen der JDK-Klassenbibliothek vorgestellt und die Klassen bei der Lösung mittelgroßer Probleme eingesetzt:

- Java-Collection-API
- IO und Streams
- GUI-Anwendungen mit JavaFX
- Parallele Programmierung mit Threads

Im dritten Teil werden weitere Programmierparadigma eingeführt und mit dem objektorientierten Paradigma verglichen:

- Funktionale Programmierung (Java-Lamdas, Haskell, Scala)
- Logische Programmierung (Prolog)

Ergänzt wird der Vorlesungsteil um einen umfassenden Übungsteil, in dem insbesondere die vermittelten Inhalte an praktischen Beispielen umgesetzt werden.

Literaturempfehlungen

Essentiell:

- Skript (wird entweder in gedruckter Form oder in elektronischer Form über das StudIP fortlaufend in der Vorlesung zur Verfügung gestellt)

Gute Sekundärliteratur:

- Heide Balzert: Lehrbuch der Objektmodellierung: Analyse und Entwurf mit der UML 2, Spektrum Akademischer Verlag
- Dietmar Ratz, Jens Scheffler, Detlev Seese, Jan Wiesenberger: Grundkurs Programmieren in Java, Carl Hanser Verlag.
- Christian Ullenboom: Java ist auch eine Insel: Programmieren lernen mit dem Standardwerk für Java-Entwickler, Rheinwerk Computing
- Christian Ullenboom: Java SE 8 Standard-Bibliothek: Das Handbuch für Entwickler, Rheinwerk Computing

Links				
Unterrichtssprache		Deutsch		
Dauer in Semestern		1 Semester		
Angebotsrhythmus Modul		SoSe		
Aufnahmekapazität Modul		unbegrenzt		
Modullevel / module level		BC (Basiscurriculum / Base curriculum)		
Modulart / typ of module		Pflicht / Mandatory		
Lehr-/Lernform / Teaching/Learning method		V+Ü		
Vorkenntnisse / Previous knowledge				
Prüfung		Prüfungszeiten		Prüfungsform
Gesamtmodul		Genauere Prüfungsformalitäten werden in der Veranstaltung bekannt gegeben.		Klausur oder mündliche Prüfung
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		4.00	SoSe	56 h
Übung		2.00	SoSe	28 h
Präsenzzeit Modul insgesamt				84 h

inf200 - Grundlagen der Technischen Informatik

Modulbezeichnung	Grundlagen der Technischen Informatik
Modulcode	inf200
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Basismodule • Fach-Bachelor Mathematik (Bachelor) > Nebenfachmodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Basismodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Wolfgang Nebel ◦ Die im Modul Lehrenden <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Wolfgang Nebel ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	Informationssysteme I
Kompetenzziele	<p>Die Studierenden verstehen den Aufbau digitaler Schaltkreise und Rechnersysteme und verfügen über Kenntnisse der grundlegenden technologischen Parameter, Kriterien, Voraussetzungen und Entwicklungen des derzeitigen und zukünftig zu erwartenden Entwurfs digitaler Hardware. Sie verstehen die Grundkonzepte aktueller Rechnerarchitekturen und des Ablaufs von Programmen hierauf. Am Ende der Veranstaltung sollen die Studierenden in der Lage sein, Rechnerarchitekturen zu analysieren, einzelne Hardwarekomponenten von Rechnern zu verstehen, sie zu entwerfen und zu optimieren sowie qualifiziert die Eigenschaften grundlegender Entwurfsalternativen zu diskutieren.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • identifizieren die wesentlichen Komponenten von digitalen Schaltkreisen und Digitalrechnern • erkennen den Wert hierarchischer und abstrakter Beschreibungen von Hardwaresystemen • benennen die grundlegenden Parameter, Kriterien, Voraussetzungen und Entwicklungen des derzeitigen und zukünftig zu erwartenden Hardware-Entwurfs basierend auf der technologischen Entwicklung • beschreiben die Grundkonzepte aktueller Rechnerarchitekturen und des Ablaufs von Programmen hierauf <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • analysieren Rechnerarchitekturen anhand einzelner Komponenten • entwerfen und optimieren einzelne Komponenten von Rechnern • transferieren systematische Methoden des Schaltkreisentwurfs auf neue Problemstellungen <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • konstruieren einfache digitale Schaltkreise in der Gruppe • vermitteln die Wirkungsweise der wesentlichen Komponenten von Digitalrechnern an andere
Modulinhalte	<p>Dieses Modul ist der erste Teil der zweisemestrigen Einführung in die Technische Informatik. Es erläutert die Konstruktionsprinzipien eines Rechners von der Ausführung eines einfachen Programms auf einer Instruction Set Architecture über die grundlegenden Techniken zur Spezifikation, Konstruktion und Optimierung der einzelnen Bestandteile eines Rechners zu den Basiskomponenten wie Gattern, Flipflops und Registern. Typische Beispiele kombinatorischer Schaltungen, wie z.B. Addierer, dienen zur Illustration modularer Entwurfstechniken. Weitergehende Entwurfstechniken werden bei sequentiellen Schaltungen, also Schaltungen mit Speicherverhalten, diskutiert und anhand von Beispielschaltungen verdeutlicht.</p>
Literaturempfehlungen	<ul style="list-style-type: none"> • Skript zur Vorlesung • Schiffmann, W.; Schmitz, R. (2001): Technische Informatik I, II, Übungsbuch; Springer Verlag, Berlin. • Dal Cin, M. (1996): Rechnerarchitektur; B.G. Teubner. • Lagemann, K. (1987): Rechnerstrukturen; Springer-Verlag, Berlin. • Oberschelp, W.; Vossen, G. (1989): Rechneraufbau und Rechnerstrukturen; Oldenbourg-Verlag. • Mano, Morris M. (1993): Computer System Architecture 3; Prentice Hall. • Gajski, D. (1997): Principles of Digital Design; Prentice Hall.

- Patterson, D.A.; Hennessy, J.L. (1997): Computer Organization and Design: The Hardware/Software Interface; 2. Edition; Morgan Kaufmann Publishers.
- Wilkinson, B. (1996): Computer Architecture Design and Performance; 2. Edition; Prentice Hall.
- Tanenbaum, A.S.(1999): Structured Computer Organization; 4. Edition; Prentice Hall.

Links				
Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	jährlich			
Aufnahmekapazität Modul	unbegrenzt			
Modullevel / module level	BC (Basiscurriculum / Base curriculum)			
Modulart / typ of module	Pflicht o. Wahlpflicht / compulsory or optioal			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge	Grundlagen der Statistik			
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul				Klausur oder mündliche Prüfung
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Übung oder Tutorium		1.00	WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf700 - Didaktik der Informatik I

Modulbezeichnung	Didaktik der Informatik I
Modulcode	inf700
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft • Zwei-Fächer-Bachelor Informatik (Bachelor) > Aufbaumodule (60 KP) • Zwei-Fächer-Bachelor Informatik (Bachelor) > Basismodule
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Ira Diethelm <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Ira Diethelm ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Fachkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • charakterisieren die erlernten Konzepte und Ansätze zur Didaktik der Informatik, wie z.B. die frühen Ansätze der Schulinformatik oder das Konzept zur Informatik im Kontext • differenzieren und diskutieren didaktische Ansätze und Konzepte zur Auswahl von informatischen Inhalten für den Schulunterricht • argumentieren den allgemeinbildenden Charakter der Informatik und vergleichen die erlernten Ansätze und Konzepte zur Didaktik der Informatik und illustrieren Gemeinsamkeiten und Widersprüche • sind in der Lage Themen für den Informatik-Unterricht anhand der erlernten Konzepte und Ansätze zur Didaktik der Informatik zu reflektieren <p>Methodenkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • vernetzen die Konzepte und Ansätze der Didaktik der Informatik mit Hilfe der didaktischen Rekonstruktion • klassifizieren die Gemeinsamkeiten und Unterschiede der Ansätze und Konzepte der Didaktik der Informatik <p>Sozialkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • diskutieren die Ansätze und Konzepte der Didaktik der Informatik mit Kommilitonen • akzeptieren Meinungen anderer und nehmen sachliche Kritik an • äußern konstruktive Kritik <p>Selbstkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • beziehen die Ansätze und Konzepte der Didaktik der Informatik in ihr Handeln ein • reflektieren ihr Selbstbild unter den Gesichtspunkten der Ansätze und Konzept der Didaktik der Informatik
Modulinhalte	<p>In der Veranstaltung wird in das Fachgebiet Didaktik der Informatik eingeführt. Dabei werden verschiedene Konzepte und Ansätze zur Didaktik der Informatik vorgestellt. Inhalte sind:</p> <ul style="list-style-type: none"> • frühe Konzepte des Informatik-Unterrichts • Allgemeinbildung und Informatik-Unterricht • der Ideenorientierte Ansatz • der Informationszentrierte Ansatz • Grundschulinformatik • der Systemorientierte Ansatz <p>Darüber hinaus werden zentrale Themen, wie zum Beispiel: "Projekte im Informatikunterricht", aufgegriffen.</p>
Literaturempfehlungen	

- Schwill, A.; Schubert, S.: Didaktik der Informatik. Berlin: Spektrum Akademischer Verlag, 2004
- Hubwieser, P.: Didaktik der Informatik. Berlin: Springer Verlag, 2000

Links				
Unterrichtssprache		Deutsch		
Dauer in Semestern		1 Semester		
Angebotsrhythmus Modul		jährlich		
Aufnahmekapazität Modul		unbegrenzt		
Modullevel / module level		AS (Akzentsetzung / Accentuation)		
Modulart / typ of module		je nach Studiengang Pflicht oder Wahlpflicht		
Lehr-/Lernform / Teaching/Learning method		V+Ü		
Vorkenntnisse / Previous knowledge		Fachliche Grundkenntnisse der Informatik		
Prüfung		Prüfungszeiten		Prüfungsform
Gesamtmodul		Am Ende der Vorlesungszeit		Mündliche Prüfung
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		2.00	SoSe	28 h
Übung		2.00	SoSe	28 h
Präsenzzeit Modul insgesamt				56 h

mat950 - Mathematik für Informatik (Diskrete Strukturen)

Modulbezeichnung	Mathematik für Informatik (Diskrete Strukturen)		
Modulcode	mat950		
Kreditpunkte	6.0 KP		
Workload	180 h		
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Aufbaumodule • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Aufbaucurriculum - Pflichtbereich • Zwei-Fächer-Bachelor Informatik (Bachelor) > Basismodule 		
Ansprechpartner/-in	Modulverantwortung <ul style="list-style-type: none"> ◦ Florian Heß ◦ Andreas Stein ◦ Sandra Stein 		
Teilnahmevoraussetzungen			
Kompetenzziele	<ul style="list-style-type: none"> • Kennenlernen und Verstehen des axiomatischen Aufbaus der Mathematik und der Bedeutung mathematischer Argumentation • Beherrschen grundlegender mathematischer Beweistechniken und deren logischer Struktur • Erkennen der Bedeutung von Voraussetzungen in mathematischen Sätzen: Lokalisierung der Voraussetzungen innerhalb der Beweise und mögliche Konsequenzen bei Wegfall von Voraussetzungen • Exemplarisches Kennenlernen weiterer mathematischer Gebiete und damit Erweiterung des eigenen mathematischen Wissens • Kennenlernen von Anwendungen • Vernetzung des eigenen mathematischen Wissens durch Herstellung von Bezügen zwischen verschiedenen mathematischen Bereichen <ul style="list-style-type: none"> • Erlernen der wesentlichen Ideen und Methoden von diskreten Strukturen in der Mathematik • Beherrschen der Grundbegriffe und wesentlichen Methoden der Graphentheorie • Beherrschen der Grundbegriffe und wesentlichen Methoden der Algebra und Zahlentheorie, wie Gruppen, Ringe, Körper, Restklassenringe, euklidischer Algorithmus, chinesischer Restsatz, Polynome • Beherrschen weiterführender Begriffe und Methoden für diskrete Strukturen, wie z.B. Primzahltests, RSA, graphentheoretische Algorithmen 		
Modulinhalte	Elemente der Aussagenlogik, Beweismethoden, Mengen, Relationen und Abbildungen, Kombinatorik, Graphen und Anwendungen, die ganzen Zahlen und ihre Restklassenringe, Gruppen und Halbgruppen		
Literaturempfehlungen	B. Kreuzler und G. Pfister: Mathematik für Informatiker, Springer-Verlag 2009 (campusweiter Online-Zugriff auf den Volltext über das Bibliothekssystem)		
Links	Weitere Literatur wird in der Veranstaltung bekannt gegeben.		
Unterrichtssprache	Deutsch		
Dauer in Semestern	1 Semester		
Angebotsrhythmus Modul	jährlich		
Aufnahmekapazität Modul	unbegrenzt		
Hinweise	Im Zwei-Fächer Bachelor Informatik ist dieses Modul im Basiscurriculum zu studieren.		
Modullevel / module level	AC (Aufbaucurriculum / Composition)		
Modulart / typ of module	Pflicht / Mandatory		
Lehr-/Lernform / Teaching/Learning method			
Vorkenntnisse / Previous knowledge			
Prüfung	Prüfungszeiten	Prüfungsform	
Gesamtmodul	Klausur nach Abschluss der Vorlesung	In diesem Modul werden Bonuspunkte erworben werden. Die Einzelheiten werden zu Beginn der Veranstaltungen mit den Studierenden besprochen und festgelegt.	
		1 Klausur (max. 3 Std.) oder 1 mündliche Prüfung (max. 30 Min.)	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus Workload Präsenzzeit

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Übung		1.00	WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

Aufbaumodule (60 KP)

inf005 - Softwaretechnik I

Modulbezeichnung	Softwaretechnik I
Modulcode	inf005
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Aufbaumodule • Fach-Bachelor Mathematik (Bachelor) > Nebenfachmodule • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Aufbaucurriculum - Pflichtbereich • Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Mastermodule • Master Umweltmodellierung (Master) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Aufbaumodule (60 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Andreas Winter <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Andreas Winter ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Ziel des Moduls ist die Vermittlung der ingenieurmäßigen Entwicklung und Wartung umfangreicher Softwaresysteme. Betrachtet wird der vollständige Software-Entwicklungsprozess inkl. Anforderungserhebung, Software-Architektur und Qualitätssicherung. Vertieft werden Grundkonzepte der objektorientierten Modellierung und Softwareentwicklung.</p> <p>Fachkompetenzen Die Studierenden: - erkennen die Phasen im Software-Lebenszyklus (vor allem Anforderungsermittlung, Entwurf, Implementierung, Qualitätssicherung) - benennen die in den Phasen anfallenden Aufgaben - wählen geeignete Methoden und Hilfsmittel in verschiedenen Phasen von Projekten aus - erkennen die Sprachmöglichkeiten der Modellierung mit UML - entwickeln und bewerten diverse Modelle in unterschiedlichen UML-Sprachen - erkennen lösen gegebene Problem mit Hilfe von Entwicklungsumgebungen</p> <p>Methodenkompetenzen Die Studierenden: - strukturieren, dokumentieren und bewerten Probleme und Lösungen mit den Werkzeugen der objekt-orientierten Modellierung - wenden Methoden und Techniken der objekt-orientierten Modellierung gezielt an</p> <p>Sozialkompetenzen Die Studierenden: - erstellen, präsentieren und diskutieren Problemlösungen mit Hilfe von Modellierungstechniken - beschreiben und lösen gegebenen Probleme der Modellierung in Gruppen</p> <p>Selbstkompetenzen Die Studierenden: - reflektieren ihr Handeln bei der Problembeschreibung und der Entwicklung von Lösungsansätzen</p>
Modulinhalte	<p>In dem Modul werden die grundlegenden Begriffe und Konzepte der Softwaretechnik vermittelt. Es sind dies u.a.:</p> <ul style="list-style-type: none"> • Notwendigkeit der Softwaretechnik • Aktivitäten und Vorgehensmodelle der Software-Entwicklung • Objektorientierte Modellierung, Metamodellierung • Synchronisation von Code und Modellen • Ermittlung von Anforderung • Definition von Software-Architekturen • Einsatz von Mustern der Software Entwicklung • Definition und Sicherung der Softwarequalität • Wartung und Betrieb von Softwaresystemen In der Übung werden Werkzeuge vorgestellt und an Beispielen eingeübt.
Literaturempfehlungen	<ol style="list-style-type: none"> 1. Folienskript zur Vorlesung 2. Ian Sommerville: Software Engineering, Addison-Wesley Longman, Amsterdam, 10. Ed. 2012 3. Jochen Ludewig, Horst Lichter: Software Engineering, dpunkt.verlag, 3. Auflage 2013 4. Helmut Balzert: Lehrbuch der Software-Technik, Spektrum Akademischer Verlag, 3. Auflage 2009 5. Chris Rupp, Stefan Queins: UML 2 glasklar. Praxiswissen für die UML-Modellierung, Carl Hanser Verlag, 4. Auflage 2012
Links	

Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	jährlich			
Aufnahmekapazität Modul	unbegrenzt			
Modullevel / module level	AC (Aufbaucurriculum / Composition)			
Modullevel / module level	---			
Modulart / typ of module	Pflicht / Mandatory			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge	- inf030 - inf031			
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Am Ende der Vorlesungszeit oder begleitend zum Veranstaltungsbetrieb (bei Portfolio)		Klausur oder mündliche Prüfung oder Portfolio	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Tutorium		2.00	WiSe	28 h
Präsenzzeit Modul insgesamt				70 h

inf700 - Didaktik der Informatik I

Modulbezeichnung	Didaktik der Informatik I
Modulcode	inf700
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Wahlbereich Informatik, Kultur und Gesellschaft • Zwei-Fächer-Bachelor Informatik (Bachelor) > Aufbaumodule (60 KP) • Zwei-Fächer-Bachelor Informatik (Bachelor) > Basismodule
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Ira Diethelm <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Ira Diethelm ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Fachkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • charakterisieren die erlernten Konzepte und Ansätze zur Didaktik der Informatik, wie z.B. die frühen Ansätze der Schulinformatik oder das Konzept zur Informatik im Kontext • differenzieren und diskutieren didaktische Ansätze und Konzepte zur Auswahl von informatischen Inhalten für den Schulunterricht • argumentieren den allgemeinbildenden Charakter der Informatik und vergleichen die erlernten Ansätze und Konzepte zur Didaktik der Informatik und illustrieren Gemeinsamkeiten und Widersprüche • sind in der Lage Themen für den Informatik-Unterricht anhand der erlernten Konzepte und Ansätze zur Didaktik der Informatik zu reflektieren <p>Methodenkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • vernetzen die Konzepte und Ansätze der Didaktik der Informatik mit Hilfe der didaktischen Rekonstruktion • klassifizieren die Gemeinsamkeiten und Unterschiede der Ansätze und Konzepte der Didaktik der Informatik <p>Sozialkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • diskutieren die Ansätze und Konzepte der Didaktik der Informatik mit Kommilitonen • akzeptieren Meinungen anderer und nehmen sachliche Kritik an • äußern konstruktive Kritik <p>Selbstkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • beziehen die Ansätze und Konzepte der Didaktik der Informatik in ihr Handeln ein • reflektieren ihr Selbstbild unter den Gesichtspunkten der Ansätze und Konzept der Didaktik der Informatik
Modulinhalte	<p>In der Veranstaltung wird in das Fachgebiet Didaktik der Informatik eingeführt. Dabei werden verschiedene Konzepte und Ansätze zur Didaktik der Informatik vorgestellt. Inhalte sind:</p> <ul style="list-style-type: none"> • frühe Konzepte des Informatik-Unterrichts • Allgemeinbildung und Informatik-Unterricht • der Ideenorientierte Ansatz • der Informationszentrierte Ansatz • Grundschulinformatik • der Systemorientierte Ansatz <p>Darüber hinaus werden zentrale Themen, wie zum Beispiel: "Projekte im Informatikunterricht", aufgegriffen.</p>
Literaturempfehlungen	

- Schwill, A.; Schubert, S.: Didaktik der Informatik. Berlin: Spektrum Akademischer Verlag, 2004
- Hubwieser, P.: Didaktik der Informatik. Berlin: Springer Verlag, 2000

Links				
Unterrichtssprache		Deutsch		
Dauer in Semestern		1 Semester		
Angebotsrhythmus Modul		jährlich		
Aufnahmekapazität Modul		unbegrenzt		
Modullevel / module level		AS (Akzentsetzung / Accentuation)		
Modulart / typ of module		je nach Studiengang Pflicht oder Wahlpflicht		
Lehr-/Lernform / Teaching/Learning method		V+Ü		
Vorkenntnisse / Previous knowledge		Fachliche Grundkenntnisse der Informatik		
Prüfung		Prüfungszeiten		Prüfungsform
Gesamtmodul		Am Ende der Vorlesungszeit		Mündliche Prüfung
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		2.00	SoSe	28 h
Übung		2.00	SoSe	28 h
Präsenzzeit Modul insgesamt				56 h

inf709 - Praktische Vertiefung der Informatik

Modulbezeichnung	Praktische Vertiefung der Informatik
Modulcode	inf709
Kreditpunkte	12.0 KP
Workload	360 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Zwei-Fächer-Bachelor Informatik (Bachelor) > Aufbaumodule (60 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Ira Diethelm <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Lehrende der Informatik <p>Modulberatung</p> <ul style="list-style-type: none"> ◦ Lehrende der Informatik
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Ziele des Moduls sind die Erweiterung und Vertiefung informatischer Grundkonzepte in praktischen Kontexten.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • analysieren und/oder implementieren Informatiksysteme bzw. Komponenten dieser in praktischen Kontexten • wenden vorhandene Grundkenntnisse auf Problemstellungen der Informatik an • konzipieren, realisieren, dokumentieren und präsentieren ein IT-Projekt <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • benennen Techniken des wissenschaftlichen Arbeitens • verfassen wissenschaftliche Texte und Präsentationen • strukturieren komplexe Problemstellungen • dokumentieren komplexe Problemstellungen und deren Lösungen geeignet <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • entwickeln Lösungen von komplexen Problemstellungen in Gruppen • bereiten Themen zielgerichtet auf und präsentieren diese in Gruppen • definieren Arbeitsschnittstellen und übernehmen Verantwortung für diese <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • charakterisieren und identifizieren Aufgaben und bearbeiten diese verbindlich • erkennen ihre Fähigkeiten und nutzen diese im Problemlöseprozess
Modulinhalte	Das Modul Praktische Vertiefung der Informatik besteht aus einem Projekt (9 KP) und einem vom Projekt unabhängigen Proseminar (3 KP) oder einem Praktikum (6 KP) und einem Proseminar (3 KP) und einem Seminar (3 KP) der Informatik.
Literaturempfehlungen	siehe dem Modul zugeordnete Veranstaltung
Links	
Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	halbjährlich
Aufnahmekapazität Modul	unbegrenzt
Modullevel / module level	AC (Aufbaucurriculum / Composition)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	1 PG, 1 S, oder 1PR, 2 SE

Vorkenntnisse / Previous knowledge

Prüfung		Prüfungszeiten		Prüfungsform	
Gesamtmodul		am Ende der Vorlesungszeit		Portfolio	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit	
Seminar und Projekt	1 Seminar und 1 Softwareprojekt	8.00	SoSe und WiSe	112 h	
Seminar und Praktikum	2 Seminare und 1 Praktikum	8.00	SoSe und WiSe	112 h	
Präsenzzeit Modul insgesamt				224 h	

inf800 - Proseminar Informatik

Modulbezeichnung	Proseminar Informatik
Modulcode	inf800
Kreditpunkte	3.0 KP
Workload	90 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Aufbaumodule • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Aufbaucurriculum - Pflichtbereich • Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Aufbaumodule (60 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Oliver Theel ◦ Die im Modul Lehrenden <p>Modulberatung</p> <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Angeleitet durch einen betreuenden Lehrenden recherchieren Studierende zu einem vorgegebenen Thema nach Literatur, arbeiten sich in diese ein, verstehen und bewerten die Quellen hinsichtlich ihrer Relevanz für das gewählte Thema, präsentieren und diskutieren ihre Erkenntnisse in einem wissenschaftlichen Vortrag und in einer nach wissenschaftlichen Standards aufgebauten Ausarbeitung.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • charakterisieren informatisches Basiswissen (Algorithmenbegriff, Datenstrukturen, Programmierung, Grundlagen der Praktischen, Technischen und Theoretischen Informatik) und wenden dieses an, • definieren und beschreiben die wesentlichen mathematischen, logischen und physikalischen Grundlagen der Informatik, • definieren und illustrieren differenziert die Kerndisziplinen der Informatik (Theoretische, Praktische und Technische Informatik), <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • begutachten Probleme, formulieren diese mit Hilfe formaler Modelle und untersuchen diese adäquat, • untersuchen Probleme anhand technischer und wissenschaftlicher Literatur, • reflektieren unter Anleitung ein wissenschaftliches Thema, verfassen angeleitet eine Seminararbeit nach wissenschaftlichen Gesichtspunkten und präsentieren ihre Ergebnisse in einem wissenschaftlichen Vortrag. <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • kommunizieren umsichtig und angemessen mit Anwendern und Fachleuten • wenden Präsentationstechniken zielgerichtet an, <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • planen ihr eigenständiges Vorgehen in der Informatik, • reflektieren ihre Beiträge kritisch und diskutieren sie mit Anwendern und Fachleuten, • ergänzen und vertiefen das im Studium erworbene Wissen selbständig und passen es den aktuellen Entwicklungen des Fachs an,
Modulinhalte	je nach zugeordneter Lehrveranstaltung
Literaturempfehlungen	
Links	
Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	halbjährlich

Aufnahmekapazität Modul	unbegrenzt	
Modullevel / module level	---	
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht	
Lehr-/Lernform / Teaching/Learning method	S	
Vorkenntnisse / Previous knowledge		
Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	Am Ende des Semesters und nach Absprache	Referat
Lehrveranstaltungsform	Seminar	
SWS	2.00	
Angebotsrhythmus	SoSe oder WiSe	
Workload Präsenzzeit	28 h	

inf401 - Grundlagen der Theoretischen Informatik

Modulbezeichnung	Grundlagen der Theoretischen Informatik
Modulcode	inf401
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Aufbaumodule • Fach-Bachelor Mathematik (Bachelor) > Nebenfachmodule • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Aufbaumodule (60 KP) • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Annegret Habel ◦ Ernst-Rüdiger Olderog <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Einführung in die Theorie der Automaten, formalen Sprachen, Berechenbarkeit und Komplexität</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • kennen verschiedene Sprachklassen (z.B. reguläre und kontextfreie Sprachen) • kennen dazugehörige Automatenmodelle (z.B. endliche Automaten, Kellerautomaten, Turingmaschinen) • erstellen Automaten, Turingmaschinen und Grammatiken zu gegebenen Aufgaben • kennen äquivalente Formalisierungen des Begriffs des Algorithmus • weisen Funktionen als algorithmisch berechenbar bzw. • Probleme als algorithmisch entscheidbar nach • kennen unentscheidbare Probleme • schätzen die Komplexität von Algorithmen ab • kennen Probleme, die deterministisch oder nichtdeterministisch in polynomieller Zeit lösbar sind <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • lernen die Mächtigkeit von abstrakten Modellen von Berechenbarkeit kennen <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • arbeiten in kleinen Gruppen an Lösungen von Aufgaben • präsentieren Lösungen von Aufgaben vor Gruppen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erlernen Ausdauer bei der Bearbeitung schwieriger Aufgaben • erlernen Präzision beim Aufschreiben von Lösungen
Modulinhalte	<p>Im ersten Teil der Vorlesung werden verschiedene Sprachklassen (reguläre und kontextfreie Sprachen) eingeführt. Für jede Sprachklasse werden die dazugehörigen Automatenmodelle (endliche Automaten und Kellerautomaten) vorgestellt, die zum Akzeptieren der jeweiligen Sprachen eingesetzt werden können. Diverse Eigenschaften der eingeführten Sprachen und Automaten werden bewiesen.</p> <p>Im zweiten Teil der Vorlesung wird untersucht, welche Funktionen algorithmisch berechenbar bzw. welche Probleme algorithmisch entscheidbar sind. Dazu wird der Begriff des Algorithmus formalisiert. Turingmaschinen und Grammatiken stellen sich als äquivalente Ansätze heraus. Es wird gezeigt, dass es Probleme gibt, die nicht algorithmisch entscheidbar sind. Dazu gehören auch viele Probleme von praktischem Interesse.</p> <p>Im dritten Teil der Vorlesung geht es um die Komplexität von Algorithmen, d.h. wie viel Zeit und Speicherplatz zum Lösen einer Aufgabe benötigt werden. Insbesondere werden Probleme betrachtet, die deterministisch oder nichtdeterministisch in polynomieller Zeit lösbar sind. Diese Problemklassen sind unter den Namen P und NP bekannt.</p>

Literaturempfehlungen

- essentiell: Skript "Grundbegriffe der Theoretischen Informatik", jeweils in aktueller Ausgabe
- empfohlen: Schöning: "Theoretische Informatik kurzgefasst", 5. Auflage, Spektrum, 2008
- Gute Sekundärliteratur: Hopcroft, Motwani, Ullman: "Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie", Pearson, 2002 (ein Klassiker...)

Links

Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt
Modullevel / module level	AC (Aufbaucurriculum / Composition)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü

Vorkenntnisse / Previous knowledge

Prüfung	Prüfungszeiten	Prüfungsform		
Gesamtmodul	Am Ende des Semesters	Klausur oder mündl. Prüfung		
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Übung oder Tutorium		1.00	WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

Praktische Vertiefung (60 KP)

inf004 - Softwareprojekt

Modulbezeichnung	Softwareprojekt
Modulcode	inf004
Kreditpunkte	9.0 KP
Workload	270 h
Verwendet in Studiengängen	

- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Biologie (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Chemie (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Comparative and European Law (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Engineering Physics (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel mehr...
- Fach-Bachelor Informatik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Interkulturelle Bildung und Beratung (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Mathematik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Nachhaltigkeitsökonomik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Pädagogik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Physik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Physik, Technik und Medizin (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Sozialwissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Umweltwissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Fach-Bachelor Wirtschaftswissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Anglistik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Biologie (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Chemie (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Elementarmathematik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Gender Studies (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Germanistik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Geschichte (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
- Zwei-Fächer-Bachelor Informatik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Kunst und Medien (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Materielle Kultur: Textil (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Mathematik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Musik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Niederlandistik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Ökonomische Bildung (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Pädagogik (Bachelor) > Praxismodule für Studierende mit außerschulischem

Berufsziel

- Zwei-Fächer-Bachelor Philosophie / Werte u. Normen (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Physik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Politik-Wirtschaft (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Slavistik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Sonderpädagogik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Sozialwissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Sportwissenschaft (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Technik (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel
- Zwei-Fächer-Bachelor Wirtschaftswissenschaften (Bachelor) > Praxismodule für Studierende mit außerschulischem Berufsziel

Ansprechpartner/-in

Modulverantwortung

- Marco Grawunder
Prüfungsberechtigt

Modulberatung

- Die im Modul Lehrenden

Teilnahmevoraussetzungen

Programmierung, Datenstrukturen und Algorithmen und Objektorientierte Modellierung und Programmierung

Kompetenzziele

Die Studierenden verfügen über praktische Kenntnisse zur Software-Entwicklung in einem iterativen Vorgehen im Team und haben dabei die wesentlichen Phasen des Software-Lebenszyklus (Anforderungen, Analyse, Entwurf, Implementierung, Test) betrachtet und regelmäßig ihre Ergebnisse präsentiert. Sie haben ihre Kenntnisse in Java vertieft.

Fachkompetenzen

Die Studierenden wenden Techniken und Methoden an und erkennen ihre Grenzen

Methodenkompetenzen Die Studierenden: - entwickeln komplexere Software mit Methoden des Software-Engineering anhand eines Vorgehensmodells und dokumentieren diese entsprechend - schätzen kleinere Aufgaben grob ab - führen einen iterativen Prozess aktiv durch - arbeiten sich selbstständig in fremde Systeme und Frameworks ein - bearbeiten komplexe Aufgaben ingenieurmäßig und nehmen eine Aufteilung in Teilaufgaben vor - organisieren kleinere Projekte und führen diese durch - geben die Ergebnisse ihrer Arbeit schriftlich und mündlich wieder

Sozialkompetenzen Die Studierenden: - arbeiten teamorientiert und lösen dabei auch Konflikte - entwickeln komplexere Software im Team und schätzen deren Aufwände in engerem Rahmen ab (Zeitmanagement) - reflektieren ihre eigene Leistung und die anderer Studierender (Review und Retrospektive)

Selbstkompetenzen Die Studierenden: - vertiefen ihre Fähigkeit zur Teamarbeit deutlich, insbesondere auch die Fähigkeit zur Konfliktlösung

Modulinhalte

Im Softwareprojekt entwickelt ein Team von Studierenden über zwei Semester ein größeres Softwaresystem. Dabei wird nach einem iterativen, Scrum-ähnlichen Prozessmodell vorgegangen, wobei typische externe Rollen von den Studierenden innerhalb des Teams wahrgenommen werden. Regelmäßige Präsentationen (pro Semester je 2 vor dem Dozenten, wöchentlich in den Tutorien) sorgen für regelmäßiges Feedback. In einem begleitenden Vorlesungsblock werden die wichtigsten Themen des Software Engineering wiederholt bzw. vertieft und für das Projekt nötige neue Methoden und Techniken bekannt gemacht

Literaturempfehlungen

Links

<https://confluence.swp.offis.uni-oldenburg.de/display/SWP>

Unterrichtssprache

Deutsch

Dauer in Semestern

2 Semester

Angebotsrhythmus Modul

jährlich

Aufnahmekapazität Modul

unbegrenzt

Modullevel / module level

Modulart / typ of module

je nach Studiengang Pflicht oder Wahlpflicht

Lehr-/Lernform / Teaching/Learning method

V + Ü + PR

Vorkenntnisse / Previous knowledge

Prüfung

Prüfungszeiten

Prüfungsform

Gesamtmodul

werden individuell festgelegt

Portfolio

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		2.00	WiSe	28 h
Übung		2.00	SoSe und WiSe	28 h
Projekt		4.00	SoSe und WiSe	56 h
Präsenzzeit Modul insgesamt				112 h

inf009 - Praktikum Datenbanken

Modulbezeichnung	Praktikum Datenbanken
Modulcode	inf009
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Marco Grawunder <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Marco Grawunder ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Ziele dieses Moduls sind die Vermittlung von praktischen Kenntnissen zu Datenbanken und Informationssystemen. Des Weiteren erlangen die Studierenden einen nachhaltigen Einblick in die technische Realisierung, Implementierung, Installation und Optimierung von Datenbankmanagementsystemen am Beispiel eines professionell eingesetzten DBS.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • verfügen über Kenntnisse zur technischen Realisierung bei der Implementierung und Programmierung von Datenbanksystemen • programmieren und implementieren datenbanknahe Systemroutinen • treffen entscheidende Vorgaben in der Modellierungsphase zur Optimierung von Datenbanksystemen • administrieren professionelle Datenbanksysteme (Installation, Verwaltung und Abstimmung) • erkennen Performance-Probleme in Datenbanksystemen und beheben diese durch entsprechende Methoden • organisieren und steuern Regelabläufe in Datenbanksystemen <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • schlagen für spezielle Anwendungsklassen konkrete Verarbeitungsprinzipien vor • reflektieren bestimmte Technologien und Vorgehensweisen bzgl. ihrer Konsequenzen <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • generieren Lösungen zu Problemen von Datenbanksystemen im Team <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen ihre Belastbarkeit bei der Implementierung und erkennen Fehler • reflektieren ihr Selbstbild und ihr Handeln
Modulinhalte	<p>Das Modul Praktikum Datenbanken ist vor allem als praktische Fortführung des Moduls Informationssysteme I konzipiert. Dieses Modul behandelt speziell technische Konzepte eines Datenbanksystems sowie praktische Ansätze in der Datenbankprogrammierung zur Lösung von Optimierungsfragen.</p> <p>Schwerpunkte sind dabei:</p> <ul style="list-style-type: none"> • Systemnahes Programmieren auf Datenbankmanagementebene • Implementierung von Teilaspekten eines Katalogsystems • Optimierungsstrategien auf Basis unterschiedlicher Anforderungen durch Parallelisierung und Partitionierung von Datenbanken

Literaturempfehlungen

- Ramez Elmasri und Shamkant B. Navathe (2007). Fundamentals of Databases Systems. Fifth Edition, Pearson/Addison Wesley
- Held Andrea (2005), Oracle 10g Hochverfügbarkeit Addison-Wesley
- Held Andrea (2015), Oracle 12c New Features Addison Wesley
- Feuerstein Steven, Pribyl Bill, Dawes Chip (2007). Oracle PL/SQL. 4. Auflage, O'Reillys Taschenbibliothek

Links	http://www-is.informatik.uni-oldenburg.de/227/	
Unterrichtssprache	Deutsch	
Dauer in Semestern	1 Semester	
Angebotsrhythmus Modul	WiSe	
Aufnahmekapazität Modul	unbegrenzt	
Modullevel / module level	AS (Akzentsetzung / Accentuation)	
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht	
Lehr-/Lernform / Teaching/Learning method	P	
Vorkenntnisse / Previous knowledge	Informationssysteme I Betriebssystemkenntnisse	
Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	Am Ende der Vorlesungszeit	Fachpraktische Übungen
Lehrveranstaltungsform	Praktikum	
SWS	4.00	
Angebotsrhythmus	WiSe	
Workload Präsenzzeit	56 h	

inf014 - Praktikum Betriebssysteme

Modulbezeichnung	Praktikum Betriebssysteme
Modulcode	inf014
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Oliver Theel <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Oliver Theel ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Ziel dieses Moduls ist es praktischen Erfahrungen bei der Analyse, beim Entwurf und bei der Implementierung von relevanten Komponenten eines Betriebssystems sowie deren Zusammenspiel miteinander zu erlernen.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • arbeiten sich in ein komplexes Softwaresystem ein • implementieren hardwarenahe Betriebssystem-Komponenten • beschreiben die Ausführung paralleler Systemoperationen • verstehen die grundlegenden Konzepte der Programmiersprache C++ • finden systematische Fehler in Software, insbesondere in paralleler Software • erarbeiten Aufgabenstellung im Team • verwenden UNIX-Standard Programme zum Erstellen von Lösungen • erkennen den Vorteil des Arbeitens mit virtuellen Maschinen <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen Herausforderungen beim Umgang mit Betriebssystemen • übertragen Realisierungskonzepte in einen praktischen Kontext • hinterfragen unterschiedliche Lösungen kritisch bzgl. ihrer Eigenschaften • wählen geeignete Lösungen zur Realisierung aus • schreiben ihre Texte mit korrekten deutschen Umlauten <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • lösen die Praktikumsaufgaben in Kleingruppen • präsentieren Lösungsvorschläge im Plenum • diskutieren ihre unterschiedlichen Lösungsvorschläge innerhalb der Kleingruppen sowie im Plenum <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • nehmen Kritik an • organisieren ihre Arbeitsabläufe innerhalb der Kleingruppe • reflektieren ihre Lösungsvorschläge unter Berücksichtigung der geäußerten Kritik • erkennen eigene Defizite bei der Umsetzung von Theorie in Praxis
Modulinhalte	<p>Das Modul vermittelt folgende Inhalte:</p> <ul style="list-style-type: none"> • Analyse eines rudimentären Betriebssystems • Konzeption und Implementierung einer Prozessverwaltung • Konzeption und Implementierung von Prozesssynchronisationsmechanismen • Konzeption und Implementierung einer virtuellen Speicherverwaltung

- Konzeption und Implementierung einer Dateiverwaltung oder Benutzerschnittstelle

Literaturempfehlungen	Patterson and Hennessy, Computer Organization and Design, 3rd edition, Morgan Kaufmann, 2007	
Links		
Unterrichtssprache	Deutsch	
Dauer in Semestern	1 Semester	
Angebotsrhythmus Modul	WiSe	
Aufnahmekapazität Modul	unbegrenzt	
Hinweise	Verknüpft mit den Modulen: <ul style="list-style-type: none"> • Betriebssysteme I • Betriebssysteme II • Verteilte Systeme 	
Modullevel / module level	AS (Akzentsetzung / Accentuation)	
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht	
Lehr-/Lernform / Teaching/Learning method	P	
Vorkenntnisse / Previous knowledge	- Betriebssysteme I - Betriebssysteme II - Programmiersprachen: C, Assembler	
Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	Am Ende des Semesters	Fachpraktische Übungen
Lehrveranstaltungsform	Praktikum	
SWS	4.00	
Angebotsrhythmus	WiSe	
Workload Präsenzzeit	56 h	

inf018 - Medienverarbeitung

Modulbezeichnung	Medienverarbeitung
Modulcode	inf018
Kreditpunkte	6.0 KP
Workload	180 h

Verwendet in Studiengängen

- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Biologie (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Chemie (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Comparative and European Law (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Engineering Physics (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer" mehr...
- Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik
- Fach-Bachelor Informatik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Interkulturelle Bildung und Beratung (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Mathematik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Nachhaltigkeitsökonomik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Pädagogik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Physik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Physik, Technik und Medizin (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Sozialwissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Umweltwissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik
- Fach-Bachelor Wirtschaftsinformatik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Fach-Bachelor Wirtschaftswissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule
- Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Mastermodule
- Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik
- Master Wirtschaftsinformatik (Master) > Akzentsetzungsmodulare der Informatik
- Zwei-Fächer-Bachelor Anglistik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Biologie (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Chemie (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Elementarmathematik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Gender Studies (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Germanistik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Geschichte (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Informatik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Kunst und Medien (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Materielle Kultur: Textil (Bachelor) > PP "Medieninformatik für Studierende"

- musisch-künstlerischer Fächer"
- Zwei-Fächer-Bachelor Mathematik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Musik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Niederlandistik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Ökonomische Bildung (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Pädagogik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Philosophie / Werte u. Normen (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Physik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Politik-Wirtschaft (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Slavistik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Sonderpädagogik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Sozialwissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Sportwissenschaft (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Technik (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"
 - Zwei-Fächer-Bachelor Wirtschaftswissenschaften (Bachelor) > PP "Medieninformatik für Studierende musisch-künstlerischer Fächer"

Ansprechpartner/-in

Modulverantwortung

◦ Susanne Boll-Westermann
Prüfungsberechtigt

◦ Susanne Boll-Westermann
◦ Die im Modul Lehrenden

Teilnahmevoraussetzungen

Kompetenzziele

Fachkompetenzen

Die Studierenden:

- benennen die Grundlagen und Eigenschaften von digitalen Medien
- benennen die zentralen Verfahren zur Kodierung und Kompression von Bildern, Video und Audio
- charakterisieren die Komplexität von Analyse, Klassifikation und Verarbeitung von unstrukturierten Medien, am Beispiel der Bildanalyse

Methodenkompetenzen

Die Studierenden:

- wenden Verfahren zur Kodierung, Kompression und Bildanalyse selbständig an

Sozialkompetenzen

Die Studierenden:

- konstruieren Lösungen in Gruppenarbeit
- präsentieren Lösungen von informatischen Problemen vor Gruppen

Selbstkompetenzen

Die Studierenden: -

Modulinhalte

Das Modul beschäftigt sich mit den Technologien der Medienverarbeitung. Ein Schwerpunkt der Vorlesung liegt auf der Kodierung von digitalen Bildern sowie Kompressionsverfahren für Bilder, Bildverbesserung und Bildverarbeitung. Die Vorlesung betrachtet weiterhin die Kodierung und Analyse von Video und Audio. Praktische Umsetzung erfolgt in Übungen anhand von kleinen Beispielen entlang den Themen der Vorlesung.

Literaturempfehlungen

- Wilhelm Burger und Mark James Burge. Digitale Bildverarbeitung: Eine Einführung mit Java und Image, J. Springer, 2006.
- Literatur im Handapparat der Abteilung in der Bibliothek. Linkliste im Lernmanagementsystem zu den einzelnen Themen der Vorlesung.

Links

<https://www.uni-oldenburg.de/informatik/medieninformatik/lehre/>

Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	WiSe			
Aufnahmekapazität Modul	unbegrenzt			
Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+P			
Vorkenntnisse / Previous knowledge				
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Zum Ende der Veranstaltungszeit		Fachpraktische Übungen	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		2.00	WiSe	28 h
Projekt		2.00	WiSe	28 h
Präsenzzeit Modul insgesamt				56 h

inf021 - Praktikum Fortgeschrittene Java-Technologien

Modulbezeichnung	Praktikum Fortgeschrittene Java-Technologien
Modulcode	inf021
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Dietrich Boles <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Dietrich Boles ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Ziel des Praktikums ist es, den Studierenden die Konzepte und Technologien der Java Plattform Standard Edition (Java SE) zu vermitteln. Die Studierenden sollen die Technologien nach dem Praktikum selbstständig bei der Entwicklung eigener großer Anwendungen einsetzen können.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen und benennen die wesentlichen Pakete der JDK-Klassenbibliothek • strukturieren größere Programme ordentlich und gestalten diese so, dass erweiterbar sind • bauen eigene Klassenbibliotheken auf • suchen selbstständig in der JDK-Klassenbibliothek nach benötigten Klassen und setzen diese zum Lösen entsprechender Probleme ein • recherchieren zur Lösung bestimmter Probleme selbstständig im Internet nach Lösungsansätzen • strukturieren ihre Programme ordentlich • verstehen und interpretieren auch größere fremde Programme • beurteilen die Qualität größerer Programme insbesondere in Bezug auf Wartbarkeit, Wiederverwendbarkeit und Erweiterbarkeit <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • recherchieren zur Lösung bestimmter Probleme selbstständig im Internet nach Lösungsansätzen <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • diskutieren mit anderen über eigene und fremde Lösungsansätze <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • reflektieren ihr Vorgehen beim Lösen von Programmierproblemen und nehmen neue Lösungsansätze, z.B. aus dem Internet, in ihr Repertoire auf
Modulinhalte	<p>Im Praktikum wird jeweils eine Auswahl folgender Themen vermittelt:</p> <ul style="list-style-type: none"> • GUIs (AWT, Swing, JavaFX) • Java-Basics und Collection-API • Grafik und Multimedia • Events • Model-View-Controller-Prinzip (MVC) • Threads • Internationalisierung und Lokalisierung • Reflection • IO, Dateien

- Tools (Compiler, Classloader, Drucker, ...)
- Speichertechnologien (XML und Serialisierung)
- Verteilte Programmierung (Sockets, RMI)
- Datenbankenzugriff (JDBC)
- Kompression
- Sicherheitskonzepte

Im Laufe des Praktikums wird durchgängig eine größere Aufgabe bearbeitet. Mit Bezug zum jeweiligen Thema der einzelnen Veranstaltungsblöcke wird diese schrittweise weiterentwickelt.

Literaturempfehlungen

- Christian Ullenboom: Java ist auch eine Insel, Rheinwerk Computing
- Christian Ullenboom: Java SE 8 Standard-Bibliothek, Rheinwerk Computing

Links	http://www.boles.de/teaching/javapraktikum/index.html	
Unterrichtssprache	Deutsch	
Dauer in Semestern	1 Semester	
Angebotsrhythmus Modul	jährlich	
Aufnahmekapazität Modul	unbegrenzt	
Modullevel / module level	AS (Akzentsetzung / Accentuation)	
Modulart / typ of module	Wahlpflicht / Elective	
Lehr-/Lernform / Teaching/Learning method	P	
Vorkenntnisse / Previous knowledge		
Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	Ende des Semesters	Fachpraktische Übungen
Lehrveranstaltungsform	Praktikum	
SWS	4.00	
Angebotsrhythmus	WiSe	
Workload Präsenzzeit	56 h	

inf202 - Praktikum Technische Informatik

Modulbezeichnung	Praktikum Technische Informatik
Modulcode	inf202
Kreditpunkte	6.0 KP
Workload	180 h

Verwendet in Studiengängen

- Fach-Bachelor Betriebswirtschaftslehre mit juristischem Schwerpunkt (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Biologie (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Chemie (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Comparative and European Law (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Engineering Physics (Bachelor) > Fachnahe Angebote Informatik mehr...
- Fach-Bachelor Informatik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Interkulturelle Bildung und Beratung (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Mathematik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Nachhaltigkeitsökonomik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Pädagogik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Pädagogisches Handeln in der Migrationsgesellschaft (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Physik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Physik, Technik und Medizin (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Sozialwissenschaften (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Umweltwissenschaften (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Fachnahe Angebote Informatik
- Fach-Bachelor Wirtschaftswissenschaften (Bachelor) > Fachnahe Angebote Informatik
- Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule
- Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik
- Zwei-Fächer-Bachelor Anglistik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Biologie (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Chemie (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Elementarmathematik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Ev. Theologie und Religionspädagogik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Gender Studies (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Germanistik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Geschichte (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Informatik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
- Zwei-Fächer-Bachelor Interdisziplinäre Sachbildung (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Kunst und Medien (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Materielle Kultur: Textil (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Mathematik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Musik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Niederlandistik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Ökonomische Bildung (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Pädagogik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Philosophie / Werte u. Normen (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Physik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Politik-Wirtschaft (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Slavistik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Sonderpädagogik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Sozialwissenschaften (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Sportwissenschaft (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Technik (Bachelor) > Fachnahe Angebote Informatik
- Zwei-Fächer-Bachelor Wirtschaftswissenschaften (Bachelor) > Fachnahe Angebote Informatik

Ansprechpartner/-in

Modulverantwortung

- Alfred Mikschl
- Die im Modul Lehrenden

Prüfungsberechtigt

- Alfred Mikschl
- Die im Modul Lehrenden

Teilnahmevoraussetzungen

Empfehlung: inf200 „Grundlagen der Technischen Informatik“

Kompetenzziele

Diese Veranstaltung versetzt die Studierenden in die Lage, informationstechnische Systeme zu analysieren, einzelne Komponenten von Rechnern zu verstehen, sie zu entwerfen und zu optimieren sowie qualifiziert über domänenspezifischen Hardwareentwurf zu diskutieren.

Fachkompetenz:

Die Studierenden

- beschreiben einzelne Komponenten von Rechnern
- entwerfen und optimieren einzelne Komponenten von Rechnern
- entwerfen und optimieren Automaten
- spezifizieren und implentieren autonome Systeme

Methodenkompetenz

Die Studierenden

- synthetisieren Rechnerarchitekturen
- können Methoden des Hardwareentwurfs auf verschiedene Systeme transferieren

Sozialkompetenz

Die Studierenden

- diskutieren qualifiziert über Hardware

Selbstkompetenz

Die Studierenden

- sind dazu in der Lage, ihren Kenntnisstand klar gegen Fachkräfte verwandter Disziplinen abzugrenzen

Modulinhalte	Dieses Modul ist der praktische Teil der Veranstaltung Einführung in die Technische Informatik	
Literaturempfehlungen	Skript zur Veranstaltung, Patterson, D.A., Hennesy, J.L.:Computer Organisation and Design: The Hardware/Software Interface	
Links		
Unterrichtssprache	Deutsch	
Dauer in Semestern	1 Semester	
Angebotsrhythmus Modul	Jedes Sommersemester	
Aufnahmekapazität Modul	unbegrenzt	
Modullevel / module level	AC (Aufbaucurriculum / Composition)	
Modulart / typ of module	Wahlpflicht / Elective	
Lehr-/Lernform / Teaching/Learning method	P	
Vorkenntnisse / Previous knowledge		
Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	Am Ende der Vorlesungszeit	Fachpraktische Übungen
Lehrveranstaltungsform	Praktikum	
SWS	4.00	
Angebotsrhythmus	SoSe	
Workload Präsenzzeit	56 h	

inf406 - Praktikum Realzeitsysteme

Modulbezeichnung	Praktikum Realzeitsysteme
Modulcode	inf406
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Ernst-Rüdiger Oldero <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Ernst-Rüdiger Oldero ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	Theoretische Informatik I und II
Kompetenzziele	<p>Praktische Erprobung von Methoden und Werkzeugen, mit denen Realzeitsysteme spezifiziert, simuliert, verifiziert und implementiert werden können. Insbesondere sollen Erfahrungen mit anfassbaren Mini-Robotern (Lego-Mindstorms) gemacht werden.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • implementieren Realzeitsysteme auf Lego-Mindstorm Robotern NXT • simulieren und verifizieren Realzeitsysteme auf der Basis von Realzeitautomaten mit dem Modelchecker UPPAAL • verwenden die Werkzeugumgebung Moby/RT zur Spezifikation und Simulation von Realzeitsysteme auf der <p>Basis von PLC-Automaten, zur Übersetzung in Java-Code für Lego-Mindstorms NXT und zur Übersetzung in UPPAAL</p> <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • realisieren Steuerungsaufgaben mit Hilfe von Lego-Mindstorms • spezifizieren Realzeitsysteme als Netzwerke von Realzeitautomaten und verifizieren diese mit UPPAAL • entwerfen Realzeitsysteme mit Hilfe von Moby/RT • realisieren systematisch anspruchsvolle zeitabhängige Steuerungsaufgaben mit Hilfe von Moby/RT, <p>Lego-Mindstorms und UPPAAL</p> <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • lösen Problemstellungen im Team • präsentieren Lösungsvorschläge und diskutieren diese <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen (Teil-)Problemstellungen von Realzeitsystemen und übernehmen Verantwortung für deren Realisierung
Modulinhalte	<p>Realzeitsysteme sind Systeme, bei denen die Zeit, zu der eine Ausgabe produziert wird oder zu der Daten gelesen werden, von Bedeutung ist. Die Modelle für Realzeitsysteme sind also gegenüber denen für herkömmliche Programmiermethoden um eine Dimension, die der Zeit, erweitert. Ein Beispiel für ein Realzeitsystem ist der Airbag im Auto, der schnell genug, aber auch nicht zu früh gezündet werden muss, weil die Wirkung des Airbags nur wenige hundertstel Sekunden nützlich ist.</p> <p>In der Veranstaltung werden Methoden und Werkzeuge vorgestellt und praktisch erprobt, mit denen Realzeitsysteme spezifiziert, simuliert, verifiziert und implementiert werden können. Insbesondere sollen Erfahrungen mit anfassbaren Mini-Robotern (Lego-Mindstorms) und zur Implementierung von</p>

Realzeitsystemen gesammelt werden.

Literaturempfehlungen	E.-R. Olderog, H. Dierks: Real-Time Systems: Formal Specification and Automatic Verification, Cambridge University Press, 2008	
Links		
Unterrichtssprache	Deutsch	
Dauer in Semestern	1 Semester	
Angebotsrhythmus Modul	unregelmäßig	
Aufnahmekapazität Modul	unbegrenzt	
Modullevel / module level	AS (Akzentsetzung / Accentuation)	
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht	
Lehr-/Lernform / Teaching/Learning method	P	
Vorkenntnisse / Previous knowledge		
Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	Am Ende der Vorlesungszeit	Fachpraktische Übungen
Lehrveranstaltungsform	Praktikum	
SWS	4.00	
Angebotsrhythmus	WiSe	
Workload Präsenzzeit	56 h	

inf803 - Spezielle Themen der Informatik I

Modulbezeichnung	Spezielle Themen der Informatik I
Modulcode	inf803
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Das Modul hat zum Ziel aktuelle Entwicklungen in der Informatik in den jeweils angemessenen Lehrveranstaltungsformen in das Studium zu integrieren.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • kennen neuere technische oder wissenschaftliche Entwicklungen der Informatik • transferieren Informatik-Methoden und -Vorgehensmodelle auf die Anforderungen von IT-Anwendungsgebieten • bewerten die Möglichkeiten und Grenzen informatischer Verfahren und Werkzeuge und setzen diese sachangemessen ein <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • begutachten Probleme, formulieren diese mit Hilfe formaler Modelle und untersuchen diese adäquat • finden (einen oder mehrerer) Lösungszugänge informatischer Probleme und stellen sie dar • wählen aufgabenangemessene Werkzeuge und Methoden aus und evaluieren diese • untersuchen Probleme anhand technischer und wissenschaftlicher Literatur <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • kooperieren im Team <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • planen ihr eigenständiges Vorgehen in der Informatik
Modulinhalte	Je nach zugeordneten Lehrveranstaltungen
Literaturempfehlungen	Werden in der zugeordneten Lehrveranstaltung bekannt gegeben.
Links	
Unterrichtsprachen	Deutsch, Englisch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	halbjährlich
Aufnahmekapazität Modul	unbegrenzt
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	2 Veranstaltungen aus V, Ü, S, P, PR
Vorkenntnisse / Previous knowledge	

Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul		Portfolio oder Referat oder mündliche Prüfung oder Klausur
Lehrveranstaltungsform	VA-Auswahl	
SWS	4.00	
Angebotsrhythmus	SoSe oder WiSe	
Workload Präsenzzeit	56 h	

inf804 - Spezielle Themen der Informatik II

Modulbezeichnung	Spezielle Themen der Informatik II
Modulcode	inf804
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Das Modul hat zum Ziel aktuelle Entwicklungen in der Informatik in den jeweils angemessenen Lehrveranstaltungsformen in das Studium zu integrieren.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • kennen neuere technische oder wissenschaftliche Entwicklungen der Informatik • transferieren Informatik-Methoden und -Vorgehensmodelle auf die Anforderungen von IT-Anwendungsgebieten • bewerten die Möglichkeiten und Grenzen informatischer Verfahren und Werkzeuge und setzen diese sachangemessen ein <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • begutachten Probleme, formulieren diese mit Hilfe formaler Modelle und untersuchen diese adäquat • finden (einen oder mehrerer) Lösungszugänge informatischer Probleme und stellen sie dar • wählen aufgabenangemessene Werkzeuge und Methoden aus und evaluieren diese • untersuchen Probleme anhand technischer und wissenschaftlicher Literatur <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • kooperieren im Team <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • planen ihr eigenständiges Vorgehen in der Informatik
Modulinhalte	Je nach zugeordneten Lehrveranstaltungen
Literaturempfehlungen	Werden in der zugeordneten Lehrveranstaltung bekannt gegeben
Links	
Unterrichtsprachen	Deutsch, Englisch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	halbjährlich
Aufnahmekapazität Modul	unbegrenzt
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	2 Veranstaltungen aus V, Ü, S, P, PR
Vorkenntnisse / Previous knowledge	

Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul		Portfolio oder Referat oder mündliche Prüfung oder Klausur
Lehrveranstaltungsform	VA-Auswahl	
SWS	4.00	
Angebotsrhythmus	SoSe oder WiSe	
Workload Präsenzzeit	56 h	

inf808 - Aktuelle Themen der Informatik

Modulbezeichnung	Aktuelle Themen der Informatik
Modulcode	inf808
Kreditpunkte	3.0 KP
Workload	90 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Fach-Bachelor Wirtschaftsinformatik (Bachelor) > Akzentsetzungsbereich Praktische Informatik und Angewandte Informatik • Master of Education (Wirtschaftspädagogik) Informatik (Master of Education) > Praktische Vertiefung der Informatik • Zwei-Fächer-Bachelor Informatik (Bachelor) > Praktische Vertiefung (60 KP)
Ansprechpartner/-in	Modulverantwortung <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden Prüfungsberechtigt <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	Das Modul hat zum Ziel aktuelle Entwicklungen in der Informatik in den jeweils angemessenen Lehrveranstaltungsformen in das Studium zu integrieren. <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • kennen neuere technische oder wissenschaftliche Entwicklungen der Informatik • transferieren Informatik-Methoden und -Vorgehensmodelle auf die Anforderungen von IT-Anwendungsgebieten • bewerten die Möglichkeiten und Grenzen informatischer Verfahren und Werkzeuge und setzen diese sachangemessen ein <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • begutachten Probleme, formulieren diese mit Hilfe formaler Modelle und untersuchen diese adäquat • finden (einen oder mehrerer) Lösungszugänge informatischer Probleme und stellen sie dar • wählen aufgabenangemessene Werkzeuge und Methoden aus und evaluieren diese • untersuchen Probleme anhand technischer und wissenschaftlicher Literatur • reflektieren unter Anleitung ein wissenschaftliches Thema, verfassen angeleitet eine Seminararbeit nach wissenschaftlichen Gesichtspunkten und präsentieren ihre Ergebnisse in einem wissenschaftlichen Vortrag <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • wenden Präsentationstechniken zielgerichtet an <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • planen ihr eigenständiges Vorgehen in der Informatik • reflektieren ihre Beiträge kritisch und diskutieren sie mit Anwendern und Fachleuten • ergänzen und vertiefen das im Studium erworbene Wissen selbständig und passen es den aktuellen Entwicklungen des Fachs an
Modulinhalte	Je nach zugeordneten Lehrveranstaltungen
Literaturempfehlungen	Werden in der zugeordneten Lehrveranstaltung bekannt gegeben
Links	
Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	unregelmäßig
Aufnahmekapazität Modul	unbegrenzt

Modullevel / module level	AS (Akzentsetzung / Accentuation)	
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht	
Lehr-/Lernform / Teaching/Learning method	1 Veranstaltung aus V, Ü, S, P, PR	
Vorkenntnisse / Previous knowledge		
Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	Portfolio oder Referat oder mündliche Prüfung oder Klausur	
Lehrveranstaltungsform	VA-Auswahl	
SWS	2.00	
Angebotsrhythmus	SoSe oder WiSe	
Workload Präsenzzeit	28 h	

Wahlpflicht Theoretische Informatik (30 KP)

inf400 - Theoretische Informatik: Logik

Modulbezeichnung	Theoretische Informatik: Logik
Modulcode	inf400
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> Fach-Bachelor Informatik (Bachelor) > Basismodule Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> Annegret Habel Ernst-Rüdiger Olderog <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> Annegret Habel Ernst-Rüdiger Olderog Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Einführung in die Aussagenlogik, Prädikatenlogik, Logik-Programmierung und Temporale Logik</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> haben Kenntnisse über Syntax, Semantik und Anwendung von Aussagenlogik, Prädikatenlogik, Logik-Programmierung und Temporale Logik spezifizieren Probleme mit Hilfe von logischen Formeln lösen Fragen über aussagelogische Formeln mit Hilfe von Wahrheitstablen ziehen logische Schlüsse der Aussagen- und Prädikatenlogik mit dem Kalkül des natürlichen Schließens beantworten Anfragen an Logik-Programme mit Hilfe der SLD-Resolution können Model-Checking von Kripke-Strukturen bezüglich CTL-Formeln algorithmisch durchführen <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> erkennen Logik als ein vielseitiges Hilfsmittel in der Informatik <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> arbeiten in kleinen Gruppen an Lösungen von Aufgaben präsentieren Lösungen von Aufgaben vor Gruppen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> erlernen Ausdauer bei der Bearbeitung schwieriger Aufgaben erlernen Präzision beim Aufschreiben von Lösungen

Modulinhalte	<p>Die Vorlesung führt in die Aussagenlogik, Prädikatenlogik, Logik-Programmierung und Temporale Logik ein. Ein gutes Verständnis von Logik ist für die Informatik von zentraler Bedeutung. Dieses wird bereits durch die weite Verbreitung der logischen Formelsprache in der Informatik belegt.</p> <p>Zum Beispiel kommen einfache Boolesche Ausdrücke in jeder Programmiersprache und beim Schaltkreisentwurf vor; Horn-Klauseln werden zur Wissensrepräsentation eingesetzt; Formeln der Prädikatenlogik und Temporalen Logik werden zum Spezifizieren von Eigenschaften von Soft- und Hardware benutzt. Neuere Anwendungen wie interaktives und automatisches Beweisen sowie Logik-Programmierung und die damit verwandte Programmiersprache PROLOG unterstreichen den Werkzeugcharakter der Logik in der Informatik.</p> <p>In der Vorlesung werden Syntax, Semantik, Verfahren und Kalküle zur Überprüfung der Gültigkeit von Formeln der Aussagenlogik, Prädikatenlogik und Temporalen Logik eingeführt und an Beispielen illustriert. Zentral ist</p>
---------------------	---

der Begriff der logischen Folgerung.
Themen:

- Aussagenlogik: Syntax und Semantik, Wahrheitstafeln, natürliches Schließen
- Prädikatenlogik: Syntax und Semantik, natürliches Schließen
- Logik-Programmierung: deklarative und prozedurale Semantik, Unifikationsalgorithmus von Robinson, SLD-Resolution, PROLOG
- Temporale Logik CTL: Syntax und Semantik mittels Kripke-Strukturen, Algorithmus zum Model-Checking von CTL

Literaturempfehlungen

Essentiell:
Skript "Logik"

Empfohlen:
D. van Dalen: Logic and Structure, Fourth Edition. Springer-Verlag, 2004.

Gute Sekundärliteratur:
U. Schöning. Logik für Informatiker, Spektrum Verlag, 2000.

Links

Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt
Modullevel / module level	BC (Basiscurriculum / Base curriculum)
Modulart / typ of module	Pflicht / Mandatory
Lehr-/Lernform / Teaching/Learning method	V & Ü

Vorkenntnisse / Previous knowledge

Prüfung	Prüfungszeiten	Prüfungsform		
Gesamtmodul	Am Ende der Vorlesungszeit	Klausur oder mündliche Prüfung		
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	SoSe	42 h
Übung oder Tutorium		1.00	SoSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf401 - Grundlagen der Theoretischen Informatik

Modulbezeichnung	Grundlagen der Theoretischen Informatik
Modulcode	inf401
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Aufbaumodule • Fach-Bachelor Mathematik (Bachelor) > Nebenfachmodule • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Aufbaumodule (60 KP) • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Annegret Habel ◦ Ernst-Rüdiger Olderog <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Einführung in die Theorie der Automaten, formalen Sprachen, Berechenbarkeit und Komplexität</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • kennen verschiedene Sprachklassen (z.B. reguläre und kontextfreie Sprachen) • kennen dazugehörige Automatenmodelle (z.B. endliche Automaten, Kellerautomaten, Turingmaschinen) • erstellen Automaten, Turingmaschinen und Grammatiken zu gegebenen Aufgaben • kennen äquivalente Formalisierungen des Begriffs des Algorithmus • weisen Funktionen als algorithmisch berechenbar bzw. • Probleme als algorithmisch entscheidbar nach • kennen unentscheidbare Probleme • schätzen die Komplexität von Algorithmen ab • kennen Probleme, die deterministisch oder nichtdeterministisch in polynomieller Zeit lösbar sind <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • lernen die Mächtigkeit von abstrakten Modellen von Berechenbarkeit kennen <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • arbeiten in kleinen Gruppen an Lösungen von Aufgaben • präsentieren Lösungen von Aufgaben vor Gruppen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erlernen Ausdauer bei der Bearbeitung schwieriger Aufgaben • erlernen Präzision beim Aufschreiben von Lösungen
Modulinhalte	<p>Im ersten Teil der Vorlesung werden verschiedene Sprachklassen (reguläre und kontextfreie Sprachen) eingeführt. Für jede Sprachklasse werden die dazugehörigen Automatenmodelle (endliche Automaten und Kellerautomaten) vorgestellt, die zum Akzeptieren der jeweiligen Sprachen eingesetzt werden können. Diverse Eigenschaften der eingeführten Sprachen und Automaten werden bewiesen.</p> <p>Im zweiten Teil der Vorlesung wird untersucht, welche Funktionen algorithmisch berechenbar bzw. welche Probleme algorithmisch entscheidbar sind. Dazu wird der Begriff des Algorithmus formalisiert. Turingmaschinen und Grammatiken stellen sich als äquivalente Ansätze heraus. Es wird gezeigt, dass es Probleme gibt, die nicht algorithmisch entscheidbar sind. Dazu gehören auch viele Probleme von praktischem Interesse.</p> <p>Im dritten Teil der Vorlesung geht es um die Komplexität von Algorithmen, d.h. wie viel Zeit und Speicherplatz zum Lösen einer Aufgabe benötigt werden. Insbesondere werden Probleme betrachtet, die deterministisch oder nichtdeterministisch in polynomieller Zeit lösbar sind. Diese Problemklassen sind unter den Namen P und NP bekannt.</p>

Literaturempfehlungen

- essentiell: Skript "Grundbegriffe der Theoretischen Informatik", jeweils in aktueller Ausgabe
- empfohlen: Schöning: "Theoretische Informatik kurzgefasst", 5. Auflage, Spektrum, 2008
- Gute Sekundärliteratur: Hopcroft, Motwani, Ullman: "Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie", Pearson, 2002 (ein Klassiker...)

Links

Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt
Modullevel / module level	AC (Aufbaucurriculum / Composition)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü

Vorkenntnisse / Previous knowledge

Prüfung	Prüfungszeiten	Prüfungsform		
Gesamtmodul	Am Ende des Semesters	Klausur oder mündl. Prüfung		
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Übung oder Tutorium		1.00	WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf402 - Graphersetzungssysteme

Modulbezeichnung	Graphersetzungssysteme
Modulcode	inf402
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Annegret Habel <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Annegret Habel ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Modellierung von Systemen, Einführung in Graphtransformationssysteme, sequentielle und parallele Unabhängigkeit, Terminierung und Konfluenz.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • benennen die Grundlagen der Graphtransformationssysteme und Graphprogramme • beschreiben Graphtransformationssysteme und Graphprogramme • definieren die Berechnungsvollständigkeit von Graphprogrammen • modellieren Systeme und Systemveränderungen • beweisen die sequentielle und parallele Unabhängigkeit von Ableitungen • beweisen die Terminierung und Konfluenz von Graphtransformationssystemen <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen Graphtransformationssysteme als ein vielseitiges Hilfsmittel zur Modellierung in der Informatik <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • arbeiten in kleinen Gruppen an Lösungen von Aufgaben • präsentieren Lösungen von Aufgaben vor Gruppen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erlernen Ausdauer bei der Bearbeitung schwieriger Aufgaben • erlernen Präzision beim Aufschreiben von Lösungen
Modulinhalte	<p>Graphen werden in praktisch allen Bereichen der Informatik benutzt, um komplexe Strukturen darzustellen. Einige Beispiele hierfür sind Flussdiagramme, Schaltbilder, Record-Strukturen, Syntaxbäume sowie funktionale und logische Ausdrücke. Derartige Strukturen lassen sich durch Graphtransformationssysteme dynamisch verändern, indem Ersetzungsregeln schrittweise auf die aktuelle Struktur angewendet werden. Die Veranstaltung führt in das Thema Graphtransformationssysteme ein und behandelt Umkehrbarkeit, Einbettbarkeit und Restriktion von Ableitungen, sequentielle und parallele Unabhängigkeit, Terminierung und Konfluenz.</p>
Literaturempfehlungen	<ul style="list-style-type: none"> • Handbook of Graph Grammars and Computing by Graph Transformation, <ul style="list-style-type: none"> ◦ Vol. 1: Foundations, World Scientific, 1997. ◦ Vol. 2: Applications, Languages and Tools, World Scientific, 1999. ◦ Vol. 3: Concurrency, Parallelism, and Distribution, World Scientific, 1999. • H. Ehrig et al.: Fundamentals of Algebraic Graph Transformation. EATCS Monographs of Theoretical Computer Science, Springer, 2006.
Links	

Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	im 2-Jahres-Zyklus			
Aufnahmekapazität Modul	unbegrenzt			
Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge	inf401: Theoretische Informatik II			
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Am Ende des Semesters		Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	SoSe oder WiSe	42 h
Übung		1.00	SoSe oder WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf403 - Kryptologie

Modulbezeichnung	Kryptologie
Modulcode	inf403
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Ansprechpartner/-in	Modulverantwortung <ul style="list-style-type: none"> ◦ Elke Wilkeit Prüfungsberechtigt <ul style="list-style-type: none"> ◦ Elke Wilkeit ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Kryptologie ist als Schlüsseltechnologie für die Absicherung weltweiter Computernetze von zentraler Bedeutung. Moderne kryptographische Techniken werden unter anderem dazu benutzt, Daten geheim zu halten, Nachrichten elektronisch zu signieren, den Zugang zu Rechnernetzen zu kontrollieren, elektronische Geldgeschäfte abzusichern, Urheberrechte zu schützen. Angesichts dieser zentralen Anwendungen sollten die Anwender einschätzen können, ob die benutzten kryptographischen Methoden effizient und sicher genug sind. Dazu müssen sie nicht nur wissen, wie die kryptographischen Verfahren funktionieren, sondern sie müssen auch deren mathematische Grundlagen verstehen. Beides wird in diesem Modul erklärt.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • benennen grundlegende Begriffe der Kryptologie und erläutern sie an Beispielen • kennen einschlägige kryptographische Verfahren, wenden diese an und schätzen ihre Sicherheit ein • sind vertraut im Umgang mit mathematischen Strukturen, die kryptographischen Verfahren zugrunde liegen • implementieren kryptographische Algorithmen und beweisen deren Korrektheit und Aufwandsabschätzungen <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • bewerten die Effizienz und Sicherheit kryptographischer Methoden • erweitern ihr Wissen über Algorithmen und deren Komplexität • entwickeln ihre Fähigkeiten der Programmierung, insbesondere den Umgang mit sehr großen Zahlen • analysieren einfache Verschlüsselungen mit Hilfe bekannter und selbst gefundener Techniken <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • verwenden die Sprache der Mathematik, um in Gruppen mit unterschiedlichem Vorwissen über Problemstellungen zu diskutieren • präsentieren ihre Ideen verständlich • erweitern und verbessern eigene Ideen durch die Vorschläge ihrer Kommilitonen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • reflektieren ihr Wissen über Sicherheit in IT-Systemen • reflektieren ihr Wissen über Algorithmen und deren Komplexität • erleben, wie sich ein Wissensgebiet innerhalb kurzer Zeit entwickelt hat • entdecken neue Anwendungsmöglichkeiten mathematischer Zusammenhänge

Modulinhalte

- Mathematische Grundlagen: Ganze Zahlen; Polynome; Kongruenzen; Restklassenringe.
- Verschlüsselung.
- Wahrscheinlichkeit und perfekte Sicherheit.
- Symmetrische Verschlüsselung (DES, ÄS)
- Primzahlerzeugung.

- Public-Key-Verschlüsselung.
- Faktorisierung und diskrete Logarithmen.
- Kryptographische Hashfunktionen und digitale Signaturen.
- Identifikation und Zertifizierung.

Literaturempfehlungen

- Skript zur Vorlesung; darin und in der Vorlesung weitere Literaturhinweise.
- Als Einstimmung: Singh, Simon: Geheime Botschaften. Hanser, 1999.

Links

Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	jährlich			
Aufnahmekapazität Modul	unbegrenzt			
Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge	Grundveranstaltungen Mathematik und Informatik			
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Am Ende des Semesters		Klausur	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		2.00	WiSe	28 h
Übung		2.00	WiSe	28 h
Präsenzzeit Modul insgesamt				56 h

inf404 - Petrinetze

Modulbezeichnung	Petrinetze
Modulcode	inf404
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Eike Best <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Eike Best ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Moderne, hoch parallele Systeme sind äußerst verhaltenskomplex. Ihre Konstruktion bedarf der Visualisierung und der algorithmischen Unterstützung. Petrinetze sind ein grundlegendes, weit verbreitetes grafische Modell für die Spezifikation paralleler Systeme. Auch stellen sie flexibel nutzbare algorithmische Methoden zur Analyse solcher Systeme bereit. Thema des Moduls ist die Theorie und die Anwendung von Petrinetzen, sowohl zur Visualisierung, als auch zur Analyse und Synthese hoch paralleler Systeme.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • definieren die Grundkonzepte von Petrinetzen • klassifizieren Petrinetze anhand ihrer Eigenschaften • analysieren Petrinetze • wenden Petrinetze auf gegebene Problemstellungen an <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • können sich auf Petrinetze aufbauende Spezifikationsmethoden leichter erschließen <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • präsentieren den Veranstaltungsteilnehmern Lösungen von Übungsaufgaben <p>Selbstkompetenzen Die Studierenden:</p>
Modulinhalte	<ul style="list-style-type: none"> • Grundlagen der Petrinetze. • Sprachen von Petrinetzen. • Erreichbarkeit, Überdeckbarkeit. • Die Markierungsgleichung. • Linear-algebraische Strukturelemente. • Free-Choice-Netze. • Programmverifikation mit Fallen. • Rechnen mit Netzen. • Entfaltungen. • Höhere Netze.
Literaturempfehlungen	<p>In erster Linie:</p> <ul style="list-style-type: none"> • Das Skript zum Modul (nur in Stud.IP) <p>Sekundär:</p> <ul style="list-style-type: none"> • Priese/Wimmel: Petri Netze (Springer-Verlag, 2001)

Links

Unterrichtssprachen	Deutsch, Englisch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü

Vorkenntnisse / Previous knowledge

Prüfung	Prüfungszeiten	Prüfungsform		
Gesamtmodul	Am Ende des Semesters	Klausur oder mündliche Prüfung		
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		2.00	WiSe	28 h
Übung		2.00	WiSe	28 h
Präsenzzeit Modul insgesamt				56 h

inf405 - Algorithmische Graphentheorie

Modulbezeichnung	Algorithmische Graphentheorie
Modulcode	inf405
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)

Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Elke Wilkeit <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Elke Wilkeit ◦ Die im Modul Lehrenden
----------------------------	---

Teilnahmevoraussetzungen

Kompetenzziele	<p>Graphen sind die in der Informatik am häufigsten verwendete Abstraktion. Jedes System, welches aus diskreten Zuständen oder Objekten und Beziehungen zwischen diesen besteht, kann als Graph modelliert werden. Viele Anwendungen erfordern effiziente Algorithmen zur Verarbeitung von Graphen (Turau, 1996). In diesem Modul werden neben einschlägigen Ergebnissen der Graphentheorie vor allem algorithmische Lösungen typischer Probleme vorgestellt. Die Algorithmen werden im Hinblick auf Effizienz und Anwendbarkeit diskutiert und auch implementiert. Ein wichtiger Aspekt dieses Moduls ist es, verschiedene Herangehensweisen an Probleme zu sehen und unterschiedliche Lösungsstrategien kennenzulernen.</p>
-----------------------	---

Fachkompetenzen

Die Studierenden:

- benennen grundlegende Begriffe der Graphentheorie und Optimierung und erläutern sie an Beispielen
- benennen algorithmische Lösungen typischer Probleme mit Hilfe von Graphen
- erkennen Situationen, in denen Graphen-Algorithmen angewandt werden können
- diskutieren Algorithmen bzgl. ihrer Effizienz und Anwendbarkeit
- implementieren Graphen-Algorithmen
- kennen Beweisstrategien und können diese wiedergeben

Methodenkompetenzen

Die Studierenden:

- erweitern ihr Wissen über Algorithmen und deren Komplexität
- entwickeln ihre Fähigkeiten der Programmierung
- erweitern ihr Spektrum an Methoden der mathematischen Modellierung

Sozialkompetenzen

Die Studierenden:

- verwenden die Sprache der Mathematik, um in Gruppen mit unterschiedlichem Vorwissen über Problemstellungen zu diskutieren
- präsentieren ihre Ideen verständlich
- erweitern und verbessern eigene Ideen durch die Vorschläge ihrer Kommilitonen

Selbstkompetenzen

Die Studierenden:

- reflektieren ihr Wissen über Algorithmen und deren Komplexität
- entwickeln zu gegebenen Problemstellungen geeignete Lösungsansätze
- hinterfragen Lösungsansätze kritisch

Modulinhalte

- Bäume
- Suchverfahren in Graphen
- Färbung von Graphen
- Flüsse in Netzwerken
- Anwendungen von Netzwerkalgorithmen
- Kürzeste Wege
- Approximative Algorithmen

Literaturempfehlungen	Jungnickel, Dieter: Graphs, Networks and Algorithms. Springer, Berlin, Heidelberg, 4th edition, 2013. Als E-Book im BIS verfügbar. Eine ausführliche Literaturliste ist im Skript zur Vorlesung zu finden.			
Links				
Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	jährlich			
Aufnahmekapazität Modul	unbegrenzt			
Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge	Grundveranstaltungen Mathematik und Informatik			
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Am Ende der Vorlesungszeit		Klausur	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	SoSe	42 h
Übung		1.00	SoSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf407 - Programmverifikation

Modulbezeichnung	Programmverifikation
Modulcode	inf407
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Ernst-Rüdiger Olderog <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Ernst-Rüdiger Olderog ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Einführung in Methoden zum Nachweis der Korrektheit von sequentiellen, parallelen und verteilten Programmen.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • beschreiben operationelle Semantiken von sequentiellen, parallelen und verteilten Programmen • kennen die Konzepte der partiellen und totalen Programmkorrektheit • zeigen die Korrektheit und Vollständigkeit von Beweissystemen • stellen Ein-Ausgabe-Spezifikationen von Programmen auf • führen Korrektheitsbeweise für Programme verschiedener Klassen mit Hilfe von Beweisregeln durch • überprüfen der Interferenz- und Deadlock-Freiheit paralleler Programme • transformieren parallele und verteilte Programme in nichtdeterministische Programme <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen Korrektheit als einen wichtigen Aspekt von Programmen und Informatik-Systemen <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • arbeiten in kleinen Gruppen an Lösungen von Aufgaben • präsentieren Lösungen von Aufgaben vor Gruppen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erlernen Ausdauer bei der Bearbeitung schwieriger Aufgaben • erlernen Präzision bei der Spezifikation von Problemen
Modulinhalte	<p>Programmverifikation ist ein systematischer Ansatz, die Fehlerfreiheit von Programmen zu zeigen. Dazu wird bewiesen, dass ein vorgegebenes Programm bestimmte wünschenswerte Verhaltenseigenschaften besitzt. Beispielsweise sollte ein Sortierprogramm nur sortierte Felder als Ergebnis abliefern. Bei sequentiellen Programmen geht es dabei vor allem um partielle Korrektheit, Terminierung und Abwesenheit von Laufzeitfehlern. Bei parallelen Programmen sind zusätzliche Verhaltenseigenschaften wichtig: Interferenz-Freiheit, Deadlock-Freiheit und faires Ablaufverhalten.</p> <p>In der Vorlesung geht es vornehmlich um die Verifikation paralleler Programme. Dazu werden klassische Methoden der Hoareschen Logik mit neueren Techniken der Programmtransformation kombiniert. Als Vorbereitung werden zunächst sequentielle Programme behandelt.</p>
Literaturempfehlungen	<ul style="list-style-type: none"> • K.R. Apt, E.-R. Olderog, Programmverifikation, Springer-Verlag, 1994 oder die erweiterte englische Ausgabe • K.R. Apt, E.-R. Olderog, Verification of Sequential and Concurrent Programs, 2nd Edition, Springer-Verlag, 1997

Links

Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	unregelmäßig			
Aufnahmekapazität Modul	unbegrenzt			
Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge	Theoretische Informatik I und II			
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Am Ende des Semesters		Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00		42 h
Übung		1.00		14 h
Präsenzzeit Modul insgesamt				56 h

inf408 - Algorithmen zur Software-Verifikation

Modulbezeichnung	Algorithmen zur Software-Verifikation
Modulcode	inf408
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none">• Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik• Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule• Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Ansprechpartner/-in	Modulverantwortung <ul style="list-style-type: none">◦ Ernst-Rüdiger Oldero Prüfungsberechtigt <ul style="list-style-type: none">◦ Ernst-Rüdiger Oldero◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>In der Vorlesung werden Algorithmen vorgestellt, die eine automatische Analyse und Verifikation komplexer Strukturen ermöglichen, wie sie bei Software-Systemen vorkommen. In den Übungen werden diese Algorithmen implementiert und an Fallstudien erprobt.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none">• führen CTL-Model-Checking an Beispielen durch• konstruieren abstrakte Kripke-Strukturen an Hand vorgegebener Datenabstraktionen und führen an Beispielen Abstraktions-Verfeinerungs-Schleifen durch• charakterisieren die Konzepte der Simulation und Bisimulation• verstehen das Konzept der Abstraktion von Daten und Transitionen• beschreiben Model-Checking-Verfahren als Instanzen von Fixpunkt-Algorithmen <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none">• spezifizieren reaktive Systeme mit Hilfe von Kripke-Strukturen und CTL-Formeln• setzen Model-Checking-Verfahren in praktische Algorithmen (in Java) um <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none">• arbeiten in Kleingruppen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none">• reflektieren ihr eigenes Handeln und beziehen dabei die vorgestellten Methoden ein
Modulinhalte	<p>Software-Systeme weisen komplexe Daten- und Kontrollstrukturen sowie immer größere Zustandsräume auf, so dass sie durch Testen nur unzulänglich auf ihre Korrektheit überprüft werden können. Es ist daher ein große Herausforderung an die Informatik, automatische Methoden zur Analyse und Verifikation von Verhaltenseigenschaften für Software-System zu entwickeln. In dieser Lehrveranstaltung werden Algorithmen aus den Bereichen der Programmanalyse und des Model-Checkings vorgestellt und praktisch erprobt. Die Algorithmen verarbeiten Transitionssysteme, wie sie aus Software-Systemen entstehen, und benutzen Abstraktionstechniken für Daten und Transitionen, um die Zustandsräume analysierbar zu machen.</p> <p>Themen: Kripke-Strukturen, Transitionssysteme, temporale Logiken CTL und CTL*, Fixpunkt-Algorithmen für rekursive CTL-Operatoren, Model-Checking Algorithmus für CTL, Simulation und Bisimulation auf Kripke-Strukturen, Sätze über die Erhaltung von Eigenschaften bei (Bi-) Simulationen, existenzielle und universelle Abstraktion von Kripke-Strukturen, Abstraktions-Verfeinerungs-Schleife (CEGAR-Methode)</p>
Literaturempfehlungen	<ul style="list-style-type: none">• E.M. Clarke, O. Grumberg, and D. Peled: Model Checking. MIT Press, 2000.• F. Nielson, H.R. Nielson, and C. Hankin: Principles of Program Analysis, Springer, 2005• E.M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, Counterexample-guided abstraction refinement for symbolic model checking, Journal of the ACM 50(5) 752-794 (2003)

Links				
Unterrichtssprache		Deutsch		
Dauer in Semestern		1 Semester		
Angebotsrhythmus Modul		unregelmäßig		
Aufnahmekapazität Modul		unbegrenzt		
Modullevel / module level		AS (Akzentsetzung / Accentuation)		
Modulart / typ of module		je nach Studiengang Pflicht oder Wahlpflicht		
Lehr-/Lernform / Teaching/Learning method		V+Ü		
Vorkenntnisse / Previous knowledge		Grundveranstaltungen in Informatik und Mathematik		
Prüfung		Prüfungszeiten		Prüfungsform
Gesamtmodul		Am Ende der Vorlesungszeit		Klausur oder mündliche Prüfung
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		2.00	SoSe oder WiSe	28 h
Übung		2.00	SoSe oder WiSe	28 h
Präsenzzeit Modul insgesamt				56 h

inf409 - Formale Sprachen

Modulbezeichnung	Formale Sprachen
Modulcode	inf409
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Theoretische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Annegret Habel <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Annegret Habel ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Einführung in die Grundlagen der Syntaxanalyse und des Compilerbaus.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • benennen die Grundlagen der Syntaxanalyse und des Compilerbaus. • beschreiben die Komplexität von grundlegenden Syntaxanalysealgorithmen. • konstruieren nicht links-rekursiven Grammatiken und Grammatiken in Normalform • testen der LL(k) und LR(k)-Eigenschaft für kontextfreie Grammatiken • konstruieren LL(k)-Parsing und LR(k)-Parsing-Action und Goto Tabellen • wenden grundlegende Syntaxanalysealgorithmen an <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen Syntaxanalysealgorithmen als ein wichtiges Hilfsmittel in der Informatik <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • arbeiten in kleinen Gruppen an Lösungen von Aufgaben • präsentieren Lösungen von Aufgaben vor Gruppen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erlernen Ausdauer bei der Bearbeitung schwieriger Aufgaben • erlernen Präzision beim Aufschreiben von Lösungen
Modulinhalte	Die Veranstaltung führt in die Grundlagen der Syntaxanalyse ein und behandelt Backtrack Parsing (Top-Down & Bottom-Up Backtracking), tabellengestützte Parsing-Methoden (Cocke-Younger-Kasami & Earley) und One-Pass No Backtrack Parsing (LL(k) und LR(k)).
Literaturempfehlungen	<ul style="list-style-type: none"> • J.E. Hopcroft, J.D Ullman: Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie. Bonn, Addison-Wesley (Deutschland), 1996. • A.V. Aho, J.D. Ullman: The Theory of Parsing, Translation, and Compiling, Vol. I: Parsing, Prentice-Hall, Englewood-Cliffs, New Jersey, 1972.
Links	
Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	im 2-Jahres-Zyklus
Aufnahmekapazität Modul	unbegrenzt

Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge	Theoretische Informatik II			
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Am Ende der Vorlesungszeit		Klausur oder mündliche Prüfung.	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		2.00	SoSe	28 h
Übung		2.00	SoSe	28 h
Präsenzzeit Modul insgesamt				56 h

Wahlpflicht Technische Informatik (30 KP)

inf200 - Grundlagen der Technischen Informatik

Modulbezeichnung	Grundlagen der Technischen Informatik
Modulcode	inf200
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Basismodule • Fach-Bachelor Mathematik (Bachelor) > Nebenfachmodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Basismodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Wolfgang Nebel ◦ Die im Modul Lehrenden <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Wolfgang Nebel ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	Informationssysteme I
Kompetenzziele	<p>Die Studierenden verstehen den Aufbau digitaler Schaltkreise und Rechnersysteme und verfügen über Kenntnisse der grundlegenden technologischen Parameter, Kriterien, Voraussetzungen und Entwicklungen des derzeitigen und zukünftig zu erwartenden Entwurfs digitaler Hardware. Sie verstehen die Grundkonzepte aktueller Rechnerarchitekturen und des Ablaufs von Programmen hierauf. Am Ende der Veranstaltung sollen die Studierenden in der Lage sein, Rechnerarchitekturen zu analysieren, einzelne Hardwarekomponenten von Rechnern zu verstehen, sie zu entwerfen und zu optimieren sowie qualifiziert die Eigenschaften grundlegender Entwurfsalternativen zu diskutieren.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • identifizieren die wesentlichen Komponenten von digitalen Schaltkreisen und Digitalrechnern • erkennen den Wert hierarchischer und abstrakter Beschreibungen von Hardwaresystemen • benennen die grundlegenden Parameter, Kriterien, Voraussetzungen und Entwicklungen des derzeitigen und zukünftig zu erwartenden Hardware-Entwurfs basierend auf der technologischen Entwicklung • beschreiben die Grundkonzepte aktueller Rechnerarchitekturen und des Ablaufs von Programmen hierauf <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • analysieren Rechnerarchitekturen anhand einzelner Komponenten • entwerfen und optimieren einzelne Komponenten von Rechnern • transferieren systematische Methoden des Schaltkreisentwurfs auf neue Problemstellungen <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • konstruieren einfache digitale Schaltkreise in der Gruppe • vermitteln die Wirkungsweise der wesentlichen Komponenten von Digitalrechnern an andere
Modulinhalte	<p>Dieses Modul ist der erste Teil der zweisemestrigen Einführung in die Technische Informatik. Es erläutert die Konstruktionsprinzipien eines Rechners von der Ausführung eines einfachen Programms auf einer Instruction Set Architecture über die grundlegenden Techniken zur Spezifikation, Konstruktion und Optimierung der einzelnen Bestandteile eines Rechners zu den Basiskomponenten wie Gattern, Flipflops und Registern. Typische Beispiele kombinatorischer Schaltungen, wie z.B. Addierer, dienen zur Illustration modularer Entwurfstechniken. Weitergehende Entwurfstechniken werden bei sequentiellen Schaltungen, also Schaltungen mit Speicherhalten, diskutiert und anhand von Beispielschaltungen verdeutlicht.</p>
Literaturempfehlungen	<ul style="list-style-type: none"> • Skript zur Vorlesung • Schiffmann, W.; Schmitz, R. (2001): Technische Informatik I, II, Übungsbuch; Springer Verlag, Berlin. • Dal Cin, M. (1996): Rechnerarchitektur; B.G. Teubner.

- Lagemann, K. (1987): Rechnerstrukturen; Springer-Verlag, Berlin.
- Oberschelp, W.; Vossen, G. (1989): Rechneraufbau und Rechnerstrukturen; Oldenbourg-Verlag.
- Mano, Morris M. (1993): Computer System Architecture 3; Prentice Hall.
- Gajski, D. (1997): Principles of Digital Design; Prentice Hall.
- Patterson, D.A.; Hennessy, J.L. (1997): Computer Organization and Design: The Hardware/Software Interface; 2. Edition; Morgan Kaufmann Publishers.
- Wilkinson, B. (1996): Computer Architecture Design and Performance; 2. Edition; Prentice Hall.
- Tanenbaum, A.S. (1999): Structured Computer Organization; 4. Edition; Prentice Hall.

Links				
Unterrichtssprache		Deutsch		
Dauer in Semestern		1 Semester		
Angebotsrhythmus Modul		jährlich		
Aufnahmekapazität Modul		unbegrenzt		
Modullevel / module level		BC (Basiscurriculum / Base curriculum)		
Modulart / typ of module		Pflicht o. Wahlpflicht / compulsory or optoal		
Lehr-/Lernform / Teaching/Learning method		V+Ü		
Vorkenntnisse / Previous knowledge		Grundlagen der Statistik		
Prüfung		Prüfungszeiten		Prüfungsform
Gesamtmodul		Klausur oder mündliche Prüfung		
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Übung oder Tutorium		1.00	WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf201 - Technische Informatik

Modulbezeichnung	Technische Informatik
Modulcode	inf201
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Aufbaumodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Wolfgang Nebel Prüfungsberechtigt ◦ Wolfgang Nebel ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Die Veranstaltung versetzt die Studierenden in die Lage, Rechnerarchitekturen zu analysieren, einzelne Komponenten von Rechnern zu verstehen, sie zu entwerfen und zu optimieren sowie qualifiziert über domänenspezifischen Hardwareentwurf zu diskutieren.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • beschreiben einzelne Komponenten von Rechnern • entwerfen und optimieren einzelne Komponenten von Rechnern • verstehen Fertigungsprozesse der VSI <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • analysieren Rechnerarchitekturen • können Methoden des Schaltkreisentwurfs auf Einsatzbereiche außerhalb der Informatik transferieren <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • diskutieren qualifiziert über Hardware und Fertigungsprozesse <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • sind dazu in der Lage, ihren Kenntnisstand klar gegen Fachkräfte verwandter Disziplinen abzugrenzen
Modulinhalte	<p>Dieses Modul ist der zweite Teil der zweisemestrigen Einführung in die Technische Informatik. Im zweiten Teilmodul werden die elektrotechnischen Grundlagen der Informatik vermittelt. Dabei wird gezeigt, wie die Grundelemente eines Rechners mittels mikroelektronischer Komponenten tatsächlich entwickelt und hergestellt werden. Danach wird ein Überblick über eingebettete Systeme gegeben.</p>
Literaturempfehlungen	<ul style="list-style-type: none"> • Skript zur Vorlesung • Oberschelp, W., Vossen, G.: Rechneraufbau und Rechnerstrukturen; Oldenbourg Verlag • Gajski, D.: Principles of Digital Design; Prentice Hall 1997 • Patterson, D.A., Hennesy, J.L.: Computer Organisation and Design: The Hardware/Software Interface; 2. Edition; Morgan Kaufman Publishers, 1997 • Tannenbaum, A.S.: Structured Computer Organization ; 4. Edition; Prentice Hall, 1999 <p>Zusätzliche Literaturhinweise folgen in der Vorlesung</p>
Links	
Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt

Modullevel / module level	AC (Aufbaucurriculum / Composition)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge				
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Am Ende der Vorlesungszeit		Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Übung oder Tutorium		1.00	WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf203 - Eingebettete Systeme I

Modulbezeichnung	Eingebettete Systeme I
Modulcode	inf203
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Wolfgang Nebel ◦ Martin Georg Fränze <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Wolfgang Nebel ◦ Martin Georg Fränze ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Das Modul leistet eine Einführung in den Entwurf digitaler eingebetteter Systeme.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • benennen funktionale und nichtfunktionale Anforderungen zur Spezifikation eingebetteter System • diskutieren den Entwurfsraum und der damit verbundenen Entwurfsmethodik eingebetteter Systeme • benennen die grundlegenden Verfahren der Steuerungs- und Regelungstechnik • charakterisieren die grundlegenden Algorithmen der digitalen Signalverarbeitung <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • konstruieren mit Modellierungswerkzeugen eingebettete Systeme und Regelungssysteme • implementieren ein eingebettetes Hardware-/Software-System • analysieren verschiedene Spezifikationsprachen anhand diverser Eigenschaften <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • konstruieren Lösungen zu gegebenen Problemen in Gruppen • präsentieren Lösungen von informatischen Problemen vor Gruppen • organisieren sich zu einer Gruppe zur Lösung eines größeren Problems mit Hilfe gängiger Projektmanagementmethoden <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen die Grenzen ihrer Belastbarkeit bei der Implementierung von Systemen, bzw. Teilaspekten dieser • lösen die Übungsaufgaben eigenverantwortlich

Modulinhalte	<p>Eingebettete Systeme übernehmen komplexe Regelungs-, Steuerungs- oder Datenverarbeitungsaufgaben in technischen Systemen. Sie bilden somit ein wichtiges Wertschöpfungspotential für Produkte der Telekommunikation, der Produktionssteuerung, im Verkehrsbereich und in elektronischen Konsumgütern. Die Funktionalität eingebetteter Systeme wird durch die Integration von Prozessoren, Spezialhardware und Software realisiert. Die Problematik des Entwurfs solcher Systeme ergibt sich durch die Heterogenität der Systemarchitektur, die Komplexität der Aufgabenstellung und durch die Notwendigkeit, vielfältige technische und ökonomische Vorgaben einhalten zu müssen.</p> <p>In diesem Modul wird zunächst ein Überblick über eingebettete Systeme und den Entwurf dieser Systeme gegeben. Vor allem im Telekommunikations- und Multimedia-Bereich spielt die digitale Signalverarbeitung eine große Rolle. Aus diesem Grund werden danach die Grundlagen und einige Algorithmen der digitalen Signalverarbeitung (z. B. MPEG-Verfahren) vorgestellt. Dagegen sind für verkehrstechnische Anwendungen die Prinzipien der Steuerung und Regelung fundamental, deren Grundlagen ebenfalls dargestellt werden. Anschließend werden die Spezifikation eingebetteter Systeme und die Eigenschaften der dafür eingesetzten Sprachen behandelt. Eine Möglichkeit zur Spezifikation solcher Systeme bieten datenflussorientierte</p>
---------------------	--

graphische Modellierungssprachen, bspw. Simulink, zusammen mit kontrollflussorientierten graphischen Spezifikationen durch StateCharts, die an dieser Stelle ausführlich vorgestellt werden. Das Modul wird mit einer Behandlung der möglichen Architekturen und der Kommunikationsmodelle in eingebetteten Systemen abgeschlossen.

Theoretische sowie praktische Übungen mit dem Entwurfswerkzeug Matlab/Simulink/Stateflow begleiten die Vorlesung und bieten die Möglichkeit den Vorlesungsstoff zu vertiefen.

Literaturempfehlungen

Foliensammlung sowie:

- Harel, D.: STATECHARTS: A Visual Formalism for Complex Systems. Science of Computer Programming, 8, North-Holland, 1987, page(231-274)
- Harel D.: Naamad, A. The STATEMATE Semantics of Statecharts. ACM Trans. Software Engineering Methods, Oct 1996
- Harel, D.; Politi, M.: Modeling Reactive Systems with Statecharts: The StateMATE Approach
- Josef Hoffmann: Matlab und Simulink: Beispielorientierte Einführung in die Simulation dynamischer Systeme, Addison-Wesley, 1998, ISBN 3-8273-1077-6
- Staunstrup, J., Wolf, W. (eds.): Hardware/Software Co-Design: Principles and Practice. Kluwer Academic Publishers, 1997, ISBN 0-7923-8013-4, chapters 1, 2, (3), 4, 6, (7), (8-10)
- U. Reimers. Digitale Fernsehtechnik. 2. Aufl., Springer, 1997, ISBN 3-540-60945-8

Sekundärliteratur:

- Debardelaben, J.A.; Gadiant, A.J.: Incorporating Cost Modeling in Embedded-System Design. IEEE Design & Test, vol 13, no. 3, 1997
- De Micheli, G.; Sami, M.: Hardware-Software Co-Design. Kluwer, 1996, ISBN 0-7923-3883-9
- Gajski, D.; Vahid, F.; Narayan, S.; Gong, J.: Specification and Design of Embedded Systems. Prentice Hall, 1994, ISBN 0-13-150731-1
- T. Painter, A. Spanias. Perceptual Coding of Digital Audio. Proceedings of the IEEE, vol 88, no 4, April 2000.
- U. Freyer. DVB Digitales Fernsehen. Verlag Technik, 1997, ISBN 3-341-01192-7
- B. Friedrichs. Kanalcodierung: Grundlagen und Anwendungen in modernen Kommunikationssystemen. Springer, 1995, ISBN 3-540-58232-0
- G.C. Clark. Error-correction coding for digital communications. 3rd printing, Plenum Press, 1988, ISBN 0-306-40615-2
- Artikelserie zum MPEG-2-Standard 3/94 10/94 und das Tutorial "Digitale Bildcodierung" 1/92 1/93, beides in "Fernseh- und Kinotechnik" (BIS: Z elt ZA 1536)

Links

Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt

Hinweise	Das Modul ist für die Studierenden der Vertiefungsrichtung "Eingebettete Systeme und Mikrorobotik" als Pflichtmodul vorgesehen. Verknüpft mit den Modulen: In dem Modul "Eingebettete Systeme II" werden weitere relevante Themen, wie der Entwurfsprozess, die HW/SW-Partitionierung, die High-Level- Synthese und Hardwarebeschreibungssprachen, diskutiert. Die beiden Module Eingebettete Systeme I und II bieten Querbezüge zu den Modulen "Rechnerarchitektur", "Realzeitbetriebssysteme" und semantikorientierten Modulen der Theoretischen Informatik. Eine Vertiefung der Kenntnisse im Entwurf eingebetteter Systeme ist in den Modulen "System Level Design" und "Low Energy System Design" möglich.
-----------------	--

Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	- Grundlagen der technischen Informatik - Technische Informatik

Prüfung	Prüfungszeiten	Prüfungsform		
Gesamtmodul	Ende des Semesters	Klausur oder mündliche Prüfung		
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Übung		1.00	WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf204 - Eingebettete Systeme II

Modulbezeichnung	Eingebettete Systeme II
Modulcode	inf204
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Wolfgang Nebel ◦ Martin Georg Fränze <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Wolfgang Nebel ◦ Martin Georg Fränze ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Das Modul leistet eine Einführung in den Entwurf digitaler eingebetteter Systeme.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • benennen der Architekturen eingebetteter Systeme • benennen spezifische Hardwarekomponenten und -architekturentscheidungen, insbes. Prozessoren • charakterisieren den Entwurfsraum und die damit verbundene Entwurfsmethodik eingebetteter Systeme • dekomponieren Steuerungs- oder Regelungsaufgaben in Teilkomponenten und setzen diese auf verschiedenen Ebenen des Entwurfsraums um • partitionieren und bauen gemischte Software-/Hardwarelösungen auf • beschreiben Architekturprinzipien zur Erzielung von Fehlertoleranz • beschreiben Analysetechniken zur Bewertung von Echtzeit- und Sicherheitsanforderungen • charakterisieren die Formalien der Hardwaresynthese <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • schätzen die Konsequenzen von Entwurfsentscheidungen bzgl. Komponentenallokation und -design in Bezug auf Energieverbrauch, Performanz und Zuverlässigkeit ein • implementieren ein eingebettetes Hardware-/Software System anhand einer gegebenen Spezifikation • modellieren Hardware mit einer Hardware-Beschreibungs-Sprache • analysieren Hardware-/Software Systeme anhand von event basierter Simulation <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • konstruieren Lösungen zu gegebenen Problemen in Gruppen • präsentieren Lösungen von Informatischen Problemen vor Gruppen • organisieren sich zu einer Gruppe, zur Lösung eines größeren Problems, mithilfe von Projektmanagementmethoden <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen die Grenzen ihrer Belastbarkeit bei der Implementierung von Systemen, bzw. Teilaspekte dieser • beschäftigen sich eigenverantwortlich mit den Übungsaufgaben
Modulinhalte	<p>Eingebettete Systeme übernehmen komplexe Regelungs-, Steuerungs- oder Datenverarbeitungsaufgaben in technischen Systemen. Sie bilden somit ein wichtiges Wertschöpfungspotential für Produkte der Telekommunikation, der Produktionssteuerung, im Verkehrsbereich und in elektronischen Konsumgütern. Die Funktionalität eingebetteter Systeme wird durch die Integration von Prozessoren, Spezialhardware und Software realisiert. Die Problematik des Entwurfs solcher Systeme ergibt sich durch die Heterogenität der Systemarchitektur, die Komplexität der Aufgabenstellung und durch die Notwendigkeit, vielfältige technische und ökonomische Vorgaben einhalten zu müssen.</p>

Dieses Modul baut auf dem Modul Eingebettete Systeme I. In diesem Modul werden zunächst unterschiedliche Architekturen eingebetteter Systeme und der dort eingesetzten Prozessoren behandelt. Gegenstand des Moduls sind anschließend Methoden der Systempartitionierung und der Synthese von Hardware-Komponenten.

Theoretische sowie praktische Übungen mit Entwurfswerkzeugen, Hardwarebeschreibungssprachen und Simulationen begleiten die Vorlesung und bieten die Möglichkeit den Vorlesungsstoff zu vertiefen.

Literaturempfehlungen

Foliensammlung sowie:

- Staunstrup, J.; Wolf, W. (eds.): Hardware/Software Co-Design: Principles and Practice. Kluwer Academic Publishers, 1997, ISBN 0-7923-8013-4, chapters 1, 2, (3), 4, 6, (7), (8-10)
- Yen, Ti-Yen; Wolf, W.: Hardware-Software Co-Synthesis of Distributed Embedded Systems. Kluwer, 1996, ISBN 0-7923-9797-5

gute Sekundärliteratur:

- Peter J. Ashenden: The Designer's Guide to VHDL. Morgan Kaufmann Publishers, 2002, ISBN 1-55860-674-2
- Lehmann, G.; Wunder, B.; Selz, M.: Schaltungsdesign mit VHDL. Franzis Verlag, 1994, ISBN 3-7723-6163-3
- J. Reichardt, B. Schwarz: VHDL-Synthese, Entwurf digitaler Schaltungen und Systeme. Oldenbourg Wissenschaftsverlag, 2000, ISBN 3-486-25128-7
- Mermet, J. (ed.): Fundamentals and Standards in Hardware Description Languages. Kluwer, 1993, ISBN 0-7923-2513-3
- De Micheli, G.; Sami, M.: Hardware-Software Co-Design. Kluwer, 1996, ISBN 0-7923-3883-9
- Gajski, D.; Vahid, F.; Narayan, S.; Gong, J.: Specification and Design of Embedded Systems. Prentice Hall, 1994, ISBN 0-13-150731-1

Links

Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	jährlich			
Aufnahmekapazität Modul	unbegrenzt			
Hinweise	Das Modul ist für die Studierenden der Vertiefungsrichtung "Eingebettete Systeme und Mikrorobotik" als Pflicht-Modul vorgesehen.			
Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge				
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul			Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	SoSe	42 h
Übung		1.00	SoSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf205 - Formale Methoden Eingebetteter Systeme

Modulbezeichnung	Formale Methoden Eingebetteter Systeme
Modulcode	inf205
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Martin Georg Fränzle Prüfungsberechtigt ◦ Martin Georg Fränzle ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • beurteilen die Tragweite der mit formalen Methoden gewinnbaren Zertifikate fundiert • beurteilen die Eignung verfügbarer Verifikationswerkzeuge für eine partikuläre Fragestellung und Systemklasse • verwenden diese Werkzeuge an realen Systemen, interpretieren die erzielten Ergebnisse und verbessern in der Folge das untersuchte System zielgerichtet • bereiten Systemmodelle für automatische Analyseverfahren vor und abstrahieren bzw. kodieren diese symbolisch (oder anderweitig) entsprechend • konzipieren und implementieren eigene Verifikationsalgorithmen <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • beherrschen die mathematische Modellierung komplexer und heterogener Systeme • kennen einschlägige mathematische Modelle dynamischer Systeme und können diese auf neue Problemklassen instanzieren <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • entwickeln und implementieren im Team grundlegende Algorithmen der automatischen Verifikation • diskutieren die Vor- und Nachteile algorithmischer Alternativen und unterschiedlicher Formalisierungen <p>Selbstkompetenzen Die Studierenden: -</p>
Modulinhalte	<p>Eingebettete Computersysteme stehen in ständiger Interaktion mit ihrer Umgebung, was zu schwer vorhersehbaren Interaktionssequenzen führen kann. Dieser Umstand erschwert Konstruktion und Validation derartiger Systeme. Vergleichbar dem Einsatz statischer und materialkundlicher Modelle in der Bauwirtschaft sind deshalb formale Modelle für verschiedene Aspekte - z.B. Ausführungszeit, Energiebedarf, mögliche Systemdynamik - eingebetteter Systeme entwickelt worden. Diese stellen den jeweiligen Aspekt des Systems in geschlossener Form dar und erlauben damit die - oft vollautomatische - Herleitung von verlässlichen Kenndaten und Zertifikaten, welche für jedes beliebige Interaktionsszenario mit der Umgebung gelten. Dies steht im Gegensatz zu Methoden des Testens oder Profilings, welche nur ausgewählte Szenarien prüfen und somit nur eine begrenzte Überdeckung bieten können.</p> <p>In diesem Modul werden verschiedene derartige Modelle erklärt und Methoden zur vollautomatischen Analyse - d.h. Herleitung von Kenndaten oder Zertifikaten - oder Synthese - d.h. automatischen Erzeugung korrekter Systementwürfe - aus derartigen Modellen erläutert und in ihrer Anwendung gezeigt.</p> <p>In den Übungen besteht die Möglichkeit, die entsprechenden Kenntnisse durch Hands-on-Erfahrung mit domänentypischen Modellierungs- und Verifikationswerkzeugen zu vertiefen, sowie in einem geführten Prozess ein (kleines) vollautomatisches Verifikationswerkzeug selbst zu erstellen.</p> <p>In der Vorlesung werden die semantischen, logischen und algorithmischen Grundlagen der automatischen Analyse eingebetteter Softwaresysteme vermittelt. Die primäre Unterweisungsform ist hierbei der medial unterstützte Vortrag sowie das didaktische Frage-Antwort-Spiel, wobei als unterstützende Medien Präsentationen, Animationen und Werkzeugvorführungen dienen.</p> <p>In den Übungen wird das in der Vorlesung erworbene Wissen vertieft und praktisch umgesetzt. Hierzu werden</p>

in der ersten Semesterhälfte zweiwöchentlich Übungsaufgaben gestellt, deren Bearbeitung in Kleingruppen zur eigenverantwortlichen Prüfung des Themenverständnisses und zum partnerschaftlichen Lernen anhängt. In der zweiten Semesterhälfte wird eine ebenfalls in Kleingruppen von jeweils 3 Studierenden zu bearbeitende größere Werkzeugentwicklungsaufgabe gestellt, deren Bearbeitung die gesamte Semesterhälfte einnimmt und die Möglichkeit des projektorientierten Lernens bietet. Die Übung dient in dieser Phase der Konsultation mit den Lehrenden; insbesondere werden Lösungsansätze und Probleme vorgestellt und diskutiert.

Das Modul vermittelt einen Überblick über semantische Modelle für reaktive Systeme, Echtzeitsysteme und hybride Systeme, sowie Beispiele für entsprechende Spezifikationslogiken. Es erläutert zustandsexplorative Verifikationsverfahren sowohl expliziter wie symbolischer Form. Die erworbenen Kenntnisse können überall eingesetzt werden, wo es um die Entwicklung zuverlässiger Software- und Hardwaresysteme geht.

Literaturempfehlungen

- Michael Huth, Mark Ryan: Logic in Computer Science: Modelling and Reasoning About Systems. Cambridge University Press, 2004.
- Christel Baier, Joost-Pieter Katoen: Principles of Model Checking. MIT Press, 2008.
- Edmund M. Clarke, Orna Grumberg, Doron A. Peled: Model Checking. MIT Press, 2000.

Links

Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü

Vorkenntnisse / Previous knowledge	Fundierte Grundkenntnisse in mathematischer Logik, diskreter Mathematik, Automaten- und Berechenbarkeitstheorie, wie sie in den Modulen "Diskrete Strukturen" und "Theoretische Informatik I + II" vermittelt werden. Zudem Programmierkenntnisse, wie sie im "Programmierkurs" erworben werden. Begründung: Die in der Vorlesung vorgestellten Verfahren basieren auf einer Operationalisierung von Semantik durch Reduktion auf logische Kodierungen und mechanisierte Prüfung logischer Aussagen. Ein Verständnis dieser Inhalte sowie ihre werkzeugtechnische Umsetzung bedarf der Grundlagen aus den vorgenannten Veranstaltungen.
---	--

Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	1. Termin: Abgabe des Semesterprojekts inkl. schriftlicher Ausarbeitung eine Woche nach Ende der Vorlesungszeit; anschließend Kolloquium und Abschlussgespräch 2. Termin: Wiederholung der Abgabe des Semesterprojekts inkl. schriftlicher Ausarbeitung zwei Wochen vor Beginn des Folgesemesters anschließend Kolloquium und Abschlussgespräch	Projekt

Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Übung		1.00	WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf207 - Grundlagen der Elektrotechnik

Modulbezeichnung	Grundlagen der Elektrotechnik
Modulcode	inf207
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master Informatik (Master) > Nicht Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Andreas Hein <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Andreas Hein ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • analysieren lineare Netzwerke (Gleich- und Wechselstrom) • benennen die Grundkonzepte der Berechnung und Nutzung der Effekte von elektrischen und magnetischen Feldern • listen die Eigenschaften einfacher elektrischer Bauelemente (Zweipole) auf • berechnen Kenngrößen von einfachen elektrischen Netzwerken • setzen computergestützte Analysewerkzeuge ein • entwerfen und realisieren einfache Schaltungen <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • übertragen Berechnungsmethoden auf komplexere dynamische Systeme • erstellen Modelle elektrischer Systeme <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • präsentieren Lösungen von Problemen • vermitteln eigene Lösungen <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • reflektieren ihre Lösungskompetenz unter Einbezug der vorgestellten Verfahren und Methoden
Modulinhalte	<p>Das Modul vermittelt die folgenden Inhalte:</p> <ul style="list-style-type: none"> • Grundbegriffe (elektrische Größen und Einheiten) • Netzwerkelemente • Berechnung von linearen Gleichstromnetzwerken (Ohmsches Gesetz, Kirchhoffsche Sätze, Überlagerungsprinzip) • Größen, Berechnung und Darstellung von elektrischen und magnetischen Feldern • Bauelemente (Kondensator und Spule) • Erweiterung um zeitabhängige, periodische Größen, Zeigerdarstellung, Berechnungen mit komplexen Effektivwertzeigern
Literaturempfehlungen	<p>Essentiell:</p> <ul style="list-style-type: none"> • Folien zur Vorlesung • Albach: Grundlagen der Elektrotechnik 1 und 2. Pearson Studium, 2004. <p>Empfohlen:</p>

- Hagmann, G.: Grundlagen der Elektrotechnik. AULA-Verlag, 2002.
- Hagmann, G.: Aufgabensammlung zu den Grundlagen der Elektrotechnik. AULA-Verlag, 2002.

Links				
Unterrichtssprache		Deutsch		
Dauer in Semestern		1 Semester		
Angebotsrhythmus Modul		jährlich		
Aufnahmekapazität Modul		unbegrenzt		
Modullevel / module level		AS (Akzentsetzung / Accentuation)		
Modulart / typ of module		je nach Studiengang Pflicht oder Wahlpflicht		
Lehr-/Lernform / Teaching/Learning method		V+Ü		
Vorkenntnisse / Previous knowledge		Modul Analysis II oder Numerik		
Prüfung		Prüfungszeiten	Prüfungsform	
Gesamtmodul		Am Ende des Semesters	Semesterbegleitende fachpraktische Übung und Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	SoSe	42 h
Übung		1.00	SoSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf208 - Mikrorobotik und Mikrosystemtechnik

Modulbezeichnung	Mikrorobotik und Mikrosystemtechnik
Modulcode	inf208
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master Informatik (Master) > Nicht Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Sergej Fatikow <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Sergej Fatikow ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Die Mikrorobotik und Mikrosystemtechnik (MST) wird als eine Schlüsseltechnologie mit großem Anwendungspotential, vor allem in der Medizin-, Fertigungs-, Kommunikations-, Bio- und Umwelt- sowie Verkehrstechnik, betrachtet. Trotz des wachsenden Interesses findet man kaum eine Lehrveranstaltung, in der alle wichtigen Bestandteile dieser breitgefächerten Forschungsrichtung behandelt worden wären. Um diese Lücke zu schließen, bietet die Abteilung für Mikrorobotik und Regelungstechnik (AMiR) diese Vorlesung an. Sie soll einen Überblick über die Mikrorobotik und MST, ihre Anwendungsgebiete sowie Lösungsansätze bei der Entwicklung verschiedenartiger Mikrosysteme geben. Die Vorlesung wird durch zahlreiche Beispiele und praktische Ergebnisse veranschaulicht.</p> <p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • benennen die Ideen, Probleme und Aktivitäten der Mikrorobotik und Mikrosystemtechnik • beschreiben die Anwendungen der Mikrorobotik und MST • charakterisieren Verfahren der MST • beschreiben die Mikromechanik auf Silizium-Basis • benennen die Prinzipien von Mikrosensoren • beschreiben Beispiele von Mikrosensoren • diskutieren informationstechnische Aspekte der MST • klassifizieren Mikroroboter <p>Methodenkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen fächerübergreifende Zusammenhänge sowie funktionale Verknüpfungen von naturwissenschaftlichen und technischen Disziplinen. • erlernen die technische Abstraktion komplexer Zusammenhänge <p>Sozialkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • lösen Problemstellungen teilweise in der Gruppe. • präsentieren der Gruppe ihre Lösungsansätze <p>Selbstkompetenz Die Studierenden:</p> <ul style="list-style-type: none"> • reflektieren ihr Wissen über die technischen Informatik • erlernen die eigenständige Vertiefung ihrer fachlichen Kompetenz
Modulinhalte	<p>Mikrorobotik und MST: Ideen, Probleme, Aktivitäten; Anwendungen der Mikrorobotik und MST; Verfahren der MST; Mikromechanik auf Silizium-Basis; LIGA-Verfahren; Mikroaktoren: Prinzipien und Beispiele (elektrostatische, piezoelektrische, magnetostriktive, elektromagnetische, Formgedächtnis-, thermomechanische, elektrorheologische und andere Aktoren); Mikrosensoren: Prinzipien und Beispiele (Kraft- und Druck-, Positions- und Geschwindigkeits-, Beschleunigungs-, Bio- und chemische, Temperatur- und andere Sensoren); informationstechnische Aspekte der MST; Entwurf und Simulation in der MST; Klassifikation von Mikrorobotern; Grobpositionierung von Mikrorobotern; Feinpositionierung von Mikrorobotern; Handhabung von Mikroobjekten: Probleme und Lösungen; Mikrogreiftechniken; Mikromontage; mikroroboterbasierte</p>

Literaturempfehlungen

Essentiell:

- Vorlesungsskript in Buchform

Empfohlen:

- Fatikow, S.: Mikroroboter und Mikromontage, Teubner, Stuttgart Leipzig, 2000
- Fatikow, S./Rembold, U.: Microsystem Technology and Microrobotics, Springer, Berlin Heidelberg New York, 1997
- Menz, W. und Mohr, J.: Mikrosystemtechnik für Ingenieure, VCH, Weinheim, 1997

Gute Sekundärliteratur:

- Brück, A. und Schmidt, A.: Angewandte Mikrotechnik, Hanser, München Wien, 2001
- Ehrfeld, W. (Hrsg.): Handbuch Mikrotechnik, Hanser, München Wien, 2000
- Elbel, Th.: Mikrosensorik, Vieweg, Wiesbaden, 1996
- Fukuda, T. and Menz, W. (Eds.): Micro Mechanical Systems, Elsevier, Amsterdam, 1998
- Gardner, J.W.: Microsensors, Wiley, Chichester, 1994
- Gerlach, G. und Dötzel, W.: Grundlagen der Mikrosystemtechnik, Hanser, München Wien, 1997
- Krause, W.: Fertigung in der Feinwerk- und Mikrotechnik, Hanser, 1995
- Mescheder, U.: Mikrosystemtechnik, Teubner, Stuttgart Leipzig, 2000
- Tränkler, H.-R. und Obermeier, E. (Hrsg.): Sensortechnik, Springer, Berlin Heidelberg, 1998
- Völklein, F. und Zetterer, Th.: Einführung in die Mikrosystemtechnik, Vieweg, Wiesbaden, 2000

Links

Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt

Hinweise

Verknüpft mit dem Modul:

- Eingebettete Systeme

Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge	Analysis II oder Numerik			
Prüfung	Prüfungszeiten	Prüfungsform		
Gesamtmodul	Am Ende der Vorlesungszeit	Fachpraktische Übung und mündliche Prüfung auf Deutsch		
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		3.00	WiSe	42 h
Übung		1.00	WiSe	14 h
Präsenzzeit Modul insgesamt				56 h

inf209 - Regelungstechnik

Modulbezeichnung	Regelungstechnik
Modulcode	inf209
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master Informatik (Master) > Nicht Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Sergej Fatikow ◦ Andreas Hein <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Sergej Fatikow ◦ Andreas Hein ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • beschreiben grundsätzliche Ansätze zur Steuerung und Regelung von technischen Systemen • diskutieren die Grundkonzepte der Modellierung von Systemen und deren Kopplung mit Reglern • benennen die Methoden zur Bestimmung von Qualitätsmerkmalen von geregelten Systemen • modellieren technische Systeme mit Hilfe von Differenzialgleichungen und deren Umsetzung in Übertragungsfunktionen • entwerfen Reglerstrukturen, überprüfen deren Stabilität und bestimmen optimale Parameter ihrer Regler <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen technische Herausforderung und lösen diese durch den Transfer von Realisierungen anderer Disziplinen <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • präsentieren Lösungsansätze <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • arbeiten sich in spezifische Fragen der Entwicklung von geregelten Systemen schnell ein
Modulinhalte	<p>Das Modul vermittelt die folgenden Inhalte:</p> <ul style="list-style-type: none"> • Grundbegriffe • Analoge Übertragungsglieder: • Lineare zeitinvariante (LZI-) Glieder • Wirkungspläne • Simulation und Modellbildung • Testsignalantworten • Frequenzgang • Differentialgleichungen und Übertragungsfunktion • Stabilität • Regelstreckenarten • Reglerarten • Lineare Regelkreise: Führungs- und Störverhalten • Stabilitätskriterien • Klassische Methoden der Analyse und Synthese: • Realisierung • Computergestützte Regelung MATLAB/Simulink

Literaturempfehlungen

Essentiell:

- Foliensammlung zur Vorlesung

Empfohlen:

- Unbehauen, H.: Regelungstechnik I, Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme
- Lutz, H. und Wendt, W.: Taschenbuch der Regelungstechnik

Links

Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester
Angebotsrhythmus Modul	jährlich
Aufnahmekapazität Modul	unbegrenzt
Modullevel / module level	AS (Akzentsetzung / Accentuation)
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht
Lehr-/Lernform / Teaching/Learning method	V+Ü
Vorkenntnisse / Previous knowledge	- Differenzialgleichungen - Analysis II - Grundlagen der Elektrotechnik

Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	Am Ende der Vorlesungszeit	Klausur oder mündliche Prüfung
Lehrveranstaltungsform	Kommentar	SWS
		Angebotsrhythmus
		Workload
		Präsenzzeit
Vorlesung	3.00	WiSe
Übung	1.00	WiSe
Präsenzzeit Modul insgesamt		56 h

inf210 - Signal- und Bildverarbeitung

Modulbezeichnung	Signal- und Bildverarbeitung
Modulcode	inf210
Kreditpunkte	6.0 KP
Workload	180 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Fach-Bachelor Informatik (Bachelor) > Akzentsetzungsbereich - Wahlbereich Informatik • Master Informatik (Master) > Nicht Informatik • Master of Education (Gymnasium) Informatik (Master of Education) > Mastermodule • Zwei-Fächer-Bachelor Informatik (Bachelor) > Wahlpflicht Technische Informatik (30 KP)

Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Martin Georg Fränze ◦ Andreas Hein <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Martin Georg Fränze ◦ Andreas Hein ◦ Die im Modul Lehrenden
----------------------------	---

Teilnahmevoraussetzungen

Kompetenzziele	<p>Fachkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • benennen die Ansätze der Signal- und Bildverarbeitung in technischen Systemen • benennen die Methoden/Algorithmen zur Aufbereitung, Filterung, Klassifikation, Interpretation und Visualisierung von Signalen und Bildern • wählen Algorithmen abhängig von deren Anwendung und Anforderungen aus • schätzen Algorithmen hinsichtlich ihrer Wirksamkeit ein • entwerfen Einzelalgorithmen und Verarbeitungsketten und bestimmen deren Qualität <p>Methodenkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • arbeiten sich in spezifische Fragen der Signal- und Bildverarbeitung schnell ein • Lösungsansätze zu präsentieren <p>Sozialkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • präsentieren Lösungsansätze • erkennen technische Herausforderungen und reagieren durch Kommunikation mit anderen Disziplinen darauf <p>Selbstkompetenzen Die Studierenden:</p> <ul style="list-style-type: none"> • reflektieren ihre Lösungen und beziehen dabei die gelernten Methoden ein
-----------------------	--

Modulinhalte	<ul style="list-style-type: none"> • Grundbegriffe • Signalverarbeitung • Signalmräume und signalverarbeitende Systeme • Diskrete und kontinuierliche Signale • Kennzeichnung von Signalübertragern anhand von Testsignalen • Darstellungsbereiche und Transformationen zwischen diesen • Zeitdiskrete Systeme und Abtastung • Schätzung und Filterung • Konstruktion mit Hilfe von MATLAB • Bildverarbeitung • Einführung/Anwendungsbereiche • Funktionstransformation • Bildverbesserung/Filterung • Segmentierung • 3D-Rekonstruktion und -Visualisierung
---------------------	---

Literaturempfehlungen

Essentiell:

- Foliensammlung zur Vorlesung

Empfohlen:

- Meyer, M.; Signalverarbeitung: Analoge und digitale Signale, Systeme und Filter
- Grüningen, D. C. v.; Digitale Signalverarbeitung: mit einer Einführung in die kontinuierlichen Signale und Systeme
- Tönnies, K.; Grundlagen der Bildverarbeitung; Pearson Studium 2005
- Lehmann, Th.; Oberschelp, W.; Pelinak, E.; Peppes, R.; Bildverarbeitung in der Medizin; Springer Verlag 1997
- Handels. H.; Medizinische Bildverarbeitung; Teubner Verlag, Stuttgart - Leipzig 2000

weiterführende Literatur wird in der Vorlesung bekannt gegeben

Links

Unterrichtssprache	Deutsch			
Dauer in Semestern	1 Semester			
Angebotsrhythmus Modul	jährlich			
Aufnahmekapazität Modul	unbegrenzt			
Modullevel / module level	AS (Akzentsetzung / Accentuation)			
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht			
Lehr-/Lernform / Teaching/Learning method	V+Ü			
Vorkenntnisse / Previous knowledge	Modul math040 Analysis II b: Differentialrechnung mehrerer Variablen			
Prüfung	Prüfungszeiten		Prüfungsform	
Gesamtmodul	Am Ende des Semesters		Klausur oder mündliche Prüfung	
Lehrveranstaltungsform	Kommentar	SWS	Angebotsrhythmus	Workload Präsenzzeit
Vorlesung		2.00	WiSe	28 h
Übung		2.00	WiSe	28 h
Präsenzzeit Modul insgesamt				56 h

Abschlussmodul

bam - Bachelorarbeitsmodul

Modulbezeichnung	Bachelorarbeitsmodul
Modulcode	bam
Kreditpunkte	15.0 KP
Workload	450 h
Verwendet in Studiengängen	<ul style="list-style-type: none"> • Zwei-Fächer-Bachelor Informatik (Bachelor) > Abschlussmodul
Ansprechpartner/-in	<p>Modulverantwortung</p> <ul style="list-style-type: none"> ◦ Ira Diethelm ◦ Die im Modul Lehrenden <p>Prüfungsberechtigt</p> <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden ◦ Die Modulverantwortlichen <p>Modulberatung</p> <ul style="list-style-type: none"> ◦ Die im Modul Lehrenden
Teilnahmevoraussetzungen	
Kompetenzziele	<p>Die Studierenden zeigen, dass sie in der Lage sind, eine vertiefte wissenschaftlich orientierte Bearbeitung eines Themas der Informatik durchzuführen.</p> <p>Fachkompetenzen: Die Studierenden bewerten die Möglichkeiten und Grenzen informatischer Verfahren und Werkzeuge und setzen diese sachangemessen ein.</p> <p>Methodenkompetenzen: Die Studierenden:</p> <ul style="list-style-type: none"> • wählen aufgabenangemessene Werkzeuge und Methoden aus und evaluieren diese, • untersuchen Probleme anhand technischer und wissenschaftlicher Literatur, • führen Softwareprojekte und den Entwurf von Hardware unter Verwendung aktueller Werkzeuge der Informatik durch, • reflektieren unter Anleitung ein wissenschaftliches Thema, verfassen angeleitet einen Artikel (Seminar- oder Abschlussarbeit) nach wissenschaftlichen Gesichtspunkten und präsentieren ihre Ergebnisse in einem wissenschaftlichen Vortrag. <p>Sozialkompetenzen: Die Studierenden:</p> <ul style="list-style-type: none"> • erkennen Konflikte und lösen diese im Team, • wenden Präsentationstechniken und Projektmanagementmethoden zielgerichtet an, • identifizieren und übernehmen Verantwortung für Aufgaben, • schätzen die gesellschaftlichen Auswirkungen ihres informatischen Handelns sowie der Informationstechnologie im Allgemeinen ab und hinterfragen diese kritisch, <p>Selbstkompetenzen: Die Studierenden:</p> <ul style="list-style-type: none"> • wählen sachangemessene, auch eigene Prioritäten aus, • planen ihr eigenständiges Vorgehen in der Informatik, • ergänzen und vertiefen das im Studium erworbene Wissen selbständig und passen es den aktuellen Entwicklungen des Fachs an, • reflektieren ihre Beiträge kritisch und diskutieren sie mit Anwendern und Fachleuten.
Modulinhalte	Ein aktuelles Thema der Informatik wird mit theoretischen, wissenschaftlichen und praktischen Anteilen bearbeitet. Die Ergebnisse werden im Rahmen eines Seminars präsentiert.
Literaturempfehlungen	Nach Vorgabe, themenbezogen.
Links	Bitte in der Studienberatung informieren! Verschiedene Abteilungen haben auch Infos und Formulare auf ihrer Webseite
Unterrichtssprache	Deutsch
Dauer in Semestern	1 Semester

Angebotsrhythmus Modul	halbjährlich	
Aufnahmekapazität Modul	unbegrenzt	
Modullevel / module level	---	
Modulart / typ of module	je nach Studiengang Pflicht oder Wahlpflicht	
Lehr-/Lernform / Teaching/Learning method		
Vorkenntnisse / Previous knowledge		
Prüfung	Prüfungszeiten	Prüfungsform
Gesamtmodul	verschieden	Abschlussarbeit, Seminarvortrag
Lehrveranstaltungsform	Seminar	
SWS	2.00	
Angebotsrhythmus	SoSe und WiSe	
Workload Präsenzzeit	28 h	

