

Modulbezeichnung / Prüfungsfach Formale Modelle		Liste -	Modulnummer 8010	Sem. 1	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Übung (2 SWS)		Häufigkeit jedes Jahr	Bewertung • Klausur oder mündl. Prüfung (100% als PL)	
Aufwand • 60 h Bearbeitung (Übungsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung und Übungen) • 60 h Anwesenheit (Vorlesung und Übungen)			Dozenten Geib, Iglér		Verantwortlich Iglér
Inhaltliche Voraussetzungen • Kenntnisse über Merkmale verteilter, eingebetteter Echtzeitsysteme und Überblick über (nicht-funktionale) Qualitätsanforderungen • Grundwissen über die prinzipiellen Phasen einer Produkt- und Systementwicklung • Grundlagen über Vorgehensmodelle, Strukturmodelle und Kontextabgrenzung mittels UML und ERM					
Lernziele Formale Modelle bilden auf vielen Gebieten der Informatik die Basis für die Beschreibung grundlegender Konzepte und Problemlösungen. Besonders komplex und anspruchsvoll gestaltet sich der Prozess der Informations- und Vorgangsmodellierung (Modellbildung und Parametrierung), wenn allgemeingültige Erkenntnisse und Aussagen über das dynamische Systemverhalten unter Berücksichtigung tangierender Restriktionen und vorgegebener Randbedingungen aus der Realwelt abgeleitet werden sollen. Nach Abschluss des Moduls haben die Studierenden die Fähigkeit, • Darstellungstechniken interaktiver Produkt- und Systementwicklungen auswählen, beurteilen und anwenden zu können, • Automatenkonstrukte und Regelsysteme als Grundstrukturen informationsverarbeitender Systeme begreifen und in der Praxis heranziehen zu können sowie • anwendungsspezifische Modellierungsformen verteilter und nebenläufiger Prozesse beurteilen und umsetzen zu können. Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von formalen mathematischen Kompetenzen bei und erweitern die Methodenkompetenzen und die Analyse-, Design- und Realisierungskompetenzen.					
Inhalte Themenswerpunkt bildet die Behandlung von Darstellungs- und Modellierungstechniken innerhalb der Informatik auf Master-Ebene, welche an konkrete Aufgabenstellungen der Echtzeit-Systementwicklung (Soft- und Hardware) angebunden ist: • Formale Modelle und Beschreibungen (Vorgehensmodelle, UML, ERM, Signalfusspläne und Blockdiagramme) • Mathematische Modelle für Prüfung von Algorithmen, Testen von Programmen und Programmverifikation • Formale Entwicklung und Systemrealisierung (Entwurf von Schaltnetzen und Schaltwerken) • Modelle zur Zuverlässigkeits-, Sicherheits- und Performancebewertung • Simulation und Regelungsprozesse (Warteschlangentheorie, Zustandsmodelle, Markov-Analyse) • Nebenläufige und verteilte Echtzeitprozesse (Grundmodelle, stochastische Petrinetze, Simulationswerkzeuge)					
Literatur Kleuker, S.: Formale Modelle der Softwareentwicklung, Vieweg+Teubner, 2009 Bremaud, P.: Markov Chains, Springer, 1999 Fowler, M.: UML konzentriert, Addison-Wesley, 2004 Horvth, A.; Telek, M.; Horvath, A.: Formal Methods and Stochastic Models for Performance Evaluation, Springer, 2008 Bollig, B.: Formal Models of Communicating Systems, Springer, 2006 Cameron, I.; Hangos, K.: Process Modelling and Model Analysis, Academic Press, 2001					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Logik und Berechenbarkeit		Liste -	Modulnummer 8020	Sem. 1	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Übung (2 SWS)		Häufigkeit jedes Jahr	Bewertung • Klausur oder mündl. Prüfung (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Übungen) • 30 h Vor- und Nachbereitung (Vorlesung und Übungen) • 60 h Bearbeitung (Übungsaufgaben)			Dozenten Barth, Panitz, Reith		Verantwortlich Reith
Inhaltliche Voraussetzungen • einfache Grundbegriffe der elementaren Logik • Umgang mit den wichtigsten Beweisverfahren • sichere Beherrschung des Induktionsprinzips und der vollständigen Induktion					
Lernziele Strukturiertes Denken und Einsatz logischer Verfahren ist eine Voraussetzung bei der Entwicklung und dem Design komplexer Systeme. Nach Abschluss des Moduls sind die Studierenden in der Lage: • Aussagenlogik und Prädikatenlogik in verschiedenen Anwendungsbereichen anzuwenden • Problemstellungen (auch theoretischer Natur) zu erkennen und korrekt zu modellieren • Formale Beweise mit mathematischen Methoden führen zu können • den Umgang mit Literatur aus dem Gebiet der Logik zu beherrschen Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von formalen und mathematischen Kompetenzen bei, erweitern die Methodenkompetenzen und die Analyse-, Design- und Realisierungskompetenzen.					
Inhalte Aussagenlogik • Syntax und Semantik • Äquivalenzen • Erfüllbarkeitsproblem SAT • Normalformen (DNF und KNF) • Hornformeln • Resolution • Folgern und Schliessen Prädikatenlogik • Syntax und Semantik • Normalformen, Skolemisierung • Äquivalenzen von Formeln der PL1 • Herbrand-Universum, Herbrand-Modelle • Resolution, Unifikation • Hornklauseln Berechenbarkeit: • Äquivalenz von Berechenbarkeitsmodellen (TM, RAM und partiell-rekursive Funktionen) • Ackermann-Funktion • Entscheidbarkeit und Aufzählbarkeit • Reduzierbarkeit • Unentscheidbare Probleme • Verbindung der Prädikatenlogik zur Berechenbarkeit • Der Gödelsche Satz • Satz von Rice • Das Rekursionstheorem					
Literatur Schöning: Logik für Informatiker, Spektrum Verlag, 2000 Ebbinghaus, Flum, Thomas: Einführung in die mathematische Logik, Spektrum Verlag, 2007 Fitting: First Order Logic and Automated Theorem Proving, Springer, 1995 Hartley Rogers Jr.: Theory of Recursive Functions and Effective Computability, The MIT Press, 1987 Arnold Oberschelp: Rekursionstheorie, B.I.-Wissenschaftsverlag, 1993					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Diskrete Mathematik		Liste -	Modulnummer 8030	Sem. 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Übung (2 SWS)	Häufigkeit jedes Jahr	Bewertung • Klausur oder mündl. Prüfung (100% als PL)		
Aufwand • 60 h Anwesenheit (Vorlesung und Übungen) • 30 h Vor- und Nachbereitung (Vorlesung und Übungen) • 60 h Bearbeitung (Übungsaufgaben)			Dozenten Reith	Verantwortlich Reith	
Inhaltliche Voraussetzungen • einfache Grundbegriffe der elementaren Logik • Umgang mit den wichtigsten Beweisverfahren • sichere Beherrschung des Induktionsprinzips und der vollständigen Induktion • einfache Kenntnisse grundlegender algebraischer Strukturen					
Lernziele Diskrete Mathematik dient als Grundlage für vertiefende theoretische Überlegungen und Untersuchungen in der Informatik. Die Studierenden können nach Abschluss der Veranstaltung • wichtige mathematische Strukturen (Gruppe, Ring, Körper, Verband) sicher verwenden, • kombinatorische Methoden zur Lösung von Abzählproblemen einsetzen, • mit erzeugenden Funktionen zur Untersuchung von Folgen umgehen. Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von formalen und mathematischen Kompetenzen bei, erweitern die Methodenkompetenzen und die Analyse-, Design- und Realisierungskompetenzen.					
Inhalte • Kombinatorische Grundbegriffe: Variationen, Kombinationen (Wiederholung von Bachelor-Themen) • Permutationen, Zyklendarstellung, Transpositionen, Signatur • Elementare Gruppentheorie • Gruppendarstellungen, Fixpunkte, Bahnen, Formel von Burnside • Polyas Abzähltheorie • Elementare Zahlentheorie (Wiederholung von Bachelor-Themen), Rechnen mit Kongruenzen (teilweise Wiederholung von Bachelor-Themen) • Auflösung rekursiver Formeln, erzeugende Funktionen • Formale Potenzreihen: arithmetische Eigenschaften • Substitution von Potenzreihen • Anwendung: Umwandlung rekursiver in explizite Formeln • Asymptotische Entwicklungen					
Literatur Ronald R. Graham, Donald E. Knuth, Oren Patashnik: Concrete Mathematics, Addison-Wesley, 1994 Peter Tittmann: Einführung in die Kombinatorik, Spektrum Verlag, 2000 Werner Struckmann und Dietmar Wätjen: Mathematik für Informatiker - Grundlagen und Anwendungen, Spektrum Verlag, 2006 B.L. van der Waerden: Algebra, Band 1, Springer, 2003					
Medienform / Unterrichtssprache • Skript/Folien, Übungsblätter					

Modulbezeichnung / Prüfungsfach Seminar Wirtschaft		Liste -	Modulnummer 8040	Sem. 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Seminar (4 SWS)	Häufigkeit jedes Jahr	Bewertung • Referat und schriftliche Ausarbeitung (100% als PL)		
Aufwand • 60 h Anwesenheit (Seminaristischer Unterricht) • 60 h Vor- und Nachbereitung (incl. Literaturstudium) • 30 h Bearbeitung (Übungsaufgaben)			Dozenten Jarass	Verantwortlich Jarass	
Inhaltliche Voraussetzungen (keine)					
Lernziele Viele Informatiker benötigen im Berufsalltag spezielle Kenntnisse der Wirtschaftswissenschaften, die sie sich u.a. per Internet besorgen und aneignen müssen, um Wechselwirkungen zu Informatik-Lösungen erkennen und beurteilen zu können. • Vertiefung der fachübergreifenden Kompetenzen durch Aufbereitung des Problems für eine erfolgreiche Internetrecherche und schrittweise Eruiierung von Informationsquellen aus dem Wirtschaftsbereich (auch außerhalb des Internet), • Vertiefung der fachübergreifenden Kompetenzen durch Erkennen und Erarbeiten von Zusammenhängen zwischen Informatik-Lösungsansätzen und wirtschaftswissenschaftlichen Aspekten gegebener Problembereiche • Vertiefung der kommunikativen Kompetenzen durch knappe Präsentation der wesentlichen Ergebnisse					
Inhalte Unterstützung betriebswirtschaftlicher Prozesse durch IT, Modellbildung Vertiefung relevanter Bereiche der Betriebswirtschaftslehre auf der Basis eines computergestützten Unternehmensplanspiels: Jeweils 3 Studenten bilden ein Unternehmen (Produktion, Beschaffung, Absatz, Finanzierung, Personal), die mit einem Produkt auf einem Markt mit anderen Unternehmen konkurrieren. Das verwendete Planspiel CABA2000 wurde an der Hochschule RheinMain speziell für Ingenieurstudenten entwickelt. Das Planspiel erfordert eine außerordentlich intensive Mitarbeit der Studierenden.					
Literatur L. JARASS: Spielerhandbuch und Übungshandbuch, kostenfrei abrufbar unter www.JARASS.com , Lehre					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite (auf www.JARASS.COM abrufbar) • IT-gestützt mittels Lehr- und Übungs-CD • Handbücher • Web/Internet					

Modulbezeichnung / Prüfungsfach Fachseminar		Liste -	Modulnummer 8050	Sem. 3	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Seminar (4 SWS)		Häufigkeit jedes Semester	Bewertung • Referat und schriftliche Ausarbeitung (100% als PL)	
Aufwand • 30 h Anwesenheit (S) • 60 h Literaturstudium, Einarbeitung (Fachthema) • 60 h Ausarbeitung und Referat			Dozenten Dozenten des Studienbereichs		Verantwortlich Schwanecke
Inhaltliche Voraussetzungen (keine)					
Lernziele Selbstständiges Erarbeiten von neuen Inhalten sowie deren verständliche Darstellung sind für einen anspruchsvollen wissenschaftlichen und beruflichen Werdegang unabdingbar. Nach Abschluss des Moduls sollen die Studierenden die folgenden Kompetenzen vertieft haben: • das selbstständige wiss. Erarbeiten eines komplexen Themas anhand von verschiedenen auch widersprüchlichen Quellen • Schärfung des Urteilvermögens • Einübung kommunikativer Kompetenz • Vertiefung der informatischen Fachkompetenzen auf dem ausgewählten Themengebiet des Seminars					
Inhalte • Aktuelle Themen aus den Schwerpunkten Vernetzte Systeme und Interaktive Anwendungen • Selbstständiges Erarbeiten eines klar abgegrenzten Themas durch aktuelle Fachliteratur und andere Quellen • Feedback durch den betreuenden Dozenten und Studierende • Schriftliche Ausarbeitung • Präsentation des Themas vor einer Gruppe von Studierenden • Diskussion im Rahmen der Seminarteilnehmer und des betreuenden Dozenten					
Literatur Aktuelle Publikationen zum gewählten Themengebiet					
Medienform / Unterrichtssprache • Seminaristischer Unterricht • Präsentationsunterlagen und technische Hilfsmittel • Zusammenfassung der schriftlichen Ausarbeitungen					

Modulbezeichnung / Prüfungsfach Master-Projekt		Liste -	Modulnummer 8060	Sem. 3	Studiengang Informatik (M.Sc.)
Credits / SWS 15 cp, 450 h 8 SWS	Lehrformen • Seminar (8 SWS)		Häufigkeit jedes Semester	Bewertung • Praktische Tätigkeit, Präsentation und Fachgespräch (100% als PL)	
Aufwand • 90 h Literaturstudium, Einarbeitung • 240 h Praktische Tätigkeit (Projekt incl. Koordination) • 90 h Ausarbeitung und Präsentation • 30 h Vor- und Nachbereitung (Projekt- u. Zeitmanagement)			Dozenten Dozenten des Studienbereichs		Verantwortlich Dörner
Inhaltliche Voraussetzungen (keine)					
Lernziele Nach Absolvieren der Veranstaltung haben die Studierenden die folgenden Kenntnisse, Fertigkeiten und Kompetenzen erweitert: <ul style="list-style-type: none"> • Vertiefung der Analyse-Kompetenzen durch Bearbeitung einer wissenschaftlich relevanten und aktuellen Fragestellung einschl. einer angemessenen Literaturrecherche in wissenschaftlicher Originalliteratur und Vertiefung der kommunikativen Fertigkeiten durch Zusammenfassung und geordneter Darstellung der Ergebnisse • Vertiefung der formalen, algorithmischen, mathematischen Kompetenzen durch Modellierung von Problemaspekten und darauf aufbauend der Design-Kompetenzen zur Entwicklung von Lösungsansätzen • Vertiefung des Fachwissens in der ausgewählten Spezialisierung und Vertiefung der Methodenkompetenz bei der Ausgestaltung von Lösungsansätzen • Vertiefung der Realisierungskompetenz in einer Projektgruppe durch organisierte Implementierung und Evaluierung im Team • Vertiefung der Implementierungsfertigkeiten in aktuellen Technologien, insbesondere auch in arbeitsteiligen Prozessen • Vertiefung der kommunikativen Kompetenz und Fertigkeiten im Konfliktmanagement durch Vorstellung und Diskussion eigener Lösungsansätze im Team, Aufnahme von Kritik an eigenen Lösungsansätzen und Feedback geben zu Lösungsansätzen von anderen Teammitgliedern • Vertiefung der Projektmanagement-Kompetenz durch Projektplanung und Reviewing, Zeitmanagement, Projekt-Dokumentation, Dokumentation von Forschungsergebnissen 					
Inhalte <ul style="list-style-type: none"> • Projektmanagement (insbesondere auch Zeit- und Ressourcenmanagement) eines überschaubaren Forschungsprojekts in einer Gruppe (andere Gruppenmitglieder sind i.d.R. Kommilitonen, können aber auch z.B. wissenschaftliche Mitarbeiter sein) • Gründliche Literaturrecherche zur Themenstellung des Masterprojekts • Präsentation der Recherche-Ergebnisse in der Gruppe in Form eines State-of-the-Art-Berichts als Grundlage für die Lösungsfindung • Selbständige Entwicklung von Lösungsansätzen für die Aufgabenstellung durch Diskussion in einer Gruppe, Abwägung von Lösungsalternativen • Umsetzung/Implementierung (im Sinne eines Proof-of-Concept) unter Nutzung aktueller Technologien und Werkzeuge • Bewertung der gefundenen Lösung nach relevanten Kriterien • Ergebnissicherung in Form einer den Regeln guter wissenschaftlicher Praxis entsprechenden Dokumentation 					
Literatur Aktuelle Originalliteratur					
Medienform / Unterrichtssprache <ul style="list-style-type: none"> • Dokumentationen (Formale Modelle, Ausarbeitungen, Messungen, ...) • Projektimplementierung einschl. Quellcode • Präsentationen • Projektmanagement-Dokumentation 					

Modulbezeichnung / Prüfungsfach Master-Thesis		Liste -	Modulnummer 9040	Sem. 4	Studiengang Informatik (M.Sc.)
Credits / SWS 30 cp, 900 h 2 SWS	Lehrformen • Seminar (2 SWS) • Master-Arbeit (0 SWS)		Häufigkeit ständig	Bewertung • Kolloquium (10% als PL) • Master-Arbeit (90% als PL)	
Aufwand • 90 h Kolloquium (Teilnahme; Vorbereitung, Präsentation, Verteidigung) • 810 h Abschlussarbeit (Selbstständige, betreute Bearbeitung; Erstellung Master-Arbeit)			Dozenten Dozenten des Studienbereichs		Verantwortlich Schwanecke
Inhaltliche Voraussetzungen (keine)					
<p>Lernziele</p> <p>Die Thesis soll Absolventen als akademische Persönlichkeiten ausweisen, die offen und kritisch gegenüber innovativen Technologien und deren Anwendungen sind. Sie sind nicht nur in der Lage, aktuelle Erkenntnisse des Fachgebietes aus Forschung und Entwicklung anzuwenden, sondern sie können auch auf der Basis ihrer erworbenen Kompetenzen neue Forschungs- und Entwicklungsergebnisse gewinnen, diese nutzbringend in Lösungen umsetzen und präsentieren.</p> <p>Durch Erbringen des Moduls werden die folgenden Kompetenzen nachgewiesen:</p> <ul style="list-style-type: none"> • Kompetenz zur Analyse komplexer, evtl. unvollständiger oder widersprüchlicher Aufgabenstellungen • Kompetenz zur Entwicklung und Anwendung formaler System- und Anwendungsmodelle • Kompetenz zur Bewertung verschiedener Lösungsalternativen • Kompetenz zur Realisierung von Lösungen auf Basis aktueller Technologien • Kompetenz zur Beurteilung von Ergebnissen • Kompetenz zur Weiterentwicklung von Modellen und Technologien der Informatik im bearbeiteten Themenbereich • Erzielung eigener Forschungsergebnisse auf publikationswürdigem Niveau • Kommunikative Kompetenz durch Präsentation und Verteidigung der eigenen Arbeiten 					
<p>Inhalte</p> <ul style="list-style-type: none"> • Analyse der Aufgabenstellung, Entwicklung und Nutzung formaler Modelle, Bewertung möglicher Alternativen • Methodisch fundierter Entwurf komplexer Systeme • Entwicklung komplexer Software unter Nutzung aktueller Technologien • Nachweis funktionaler und nicht-funktionaler Eigenschaften • Wissenschaftliche Dokumentation in Form der Master-Thesis 					
<p>Literatur</p> <p>Abhängig von der Themenstellung der Thesis</p>					
<p>Medienform / Unterrichtssprache</p> <ul style="list-style-type: none"> • Master-Thesis (gedruckt, gebunden), 4 Exemplare, Datenträger • Master-Kolloquium: Präsentation, Verteidigung 					

Modulbezeichnung / Prüfungsfach Ausgewählte Themen der Informatik		Liste IM2	Modulnummer 8110	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Dozenten des Studienbereichs, Gastprofessoren, Lehrbeauftragte		Verantwortlich Schwanecke
Inhaltliche Voraussetzungen (keine)					
Lernziele Die Studierenden haben nach Abschluss des Moduls <ul style="list-style-type: none"> • spezifische Kenntnisse in den ausgewählten Themengebieten entwickelt und können entsprechende inhaltliche Zusammenhänge darlegen • ihre analytischen Fähigkeiten bei der Betrachtung komplexer Systeme erweitert • ihre formalen mathematischen und algorithmischen Fähigkeiten durch Anwendung spezifischer formaler Methoden der ausgewählten Themengebiete erweitert • ihr Beurteilungsvermögen durch Vergleich verschiedener Entwürfe und Implementierungen in dem zugehörigen Praktikum bzw. den durchgeführten Übungen gefestigt. 					
Inhalte Ausgewählte Themengebiete der Informatik auf Master-Niveau					
Literatur Abhängig von den ausgewählten Themengebieten					
Medienform / Unterrichtssprache • Abhängig von den ausgewählten Themengebieten					

Modulbezeichnung / Prüfungsfach Constraint-basierte Systeme		Liste IM2	Modulnummer 8120	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Barth, Panitz		Verantwortlich Barth
Inhaltliche Voraussetzungen • Lineare Gleichungssysteme • Laufzeit und Komplexität • Backtracking, Greedy					
Lernziele Die Studierenden sind nach Abschluss des Moduls in der Lage, • mathematische Optimierungsprobleme adäquat zu modellieren und mit Hilfe von Constraint-Lösern praktisch zu lösen, • Constraint-Löser in verschiedenen Programmierumgebungen einzusetzen, • spezifische Constraint-Solver zu entwerfen und in eine Programmierumgebung einzubetten. Die erworbenen Fähigkeiten erweitern die formalen, algorithmischen und mathematischen Kompetenzen, die Methodenkompetenzen und die Analyse-, Design- und Realisierungskompetenzen.					
Inhalte Constraints und Constraint-Systems • Erfüllbarkeit, Implikation, Projektion • Simplifikation und Darstellung eines Lösungsraums • Modellierung von Constraint-Problemen • Einbettung in Programmiersprachen, Constraint Logic Programming CLP(X), Bibliotheken Constraints über endlichen Wertebereichen, Finite Domain Constraints • Constraint Satisfaction Probleme • Konsistenzerhaltung, Relaxation • Backtracking, Labeling, Lösungsraum durchsuchen • Globale Constraints (z.B. all_different, symmetrisch) • Redundante Constraints • Optimierungsprobleme, Operations Research • Modellierung praktischer Probleme (z.B. Scheduling Probleme) Lineare arithmetische Constraints • Lineare Gleichungen und Ungleichungen • Simplex-Methode • Modellierung praktischer Probleme					
Literatur Petra Hofstedt und Armin Wolf: Einführung in die Constraint-Programmierung, Springer, 2007 Krzysztof Apt: Principles of Constraint Programming, Cambridge University Press, 2003 Pascal Van Hentenryck: Constraint Satisfaction in Logic Programming, MIT Press, 1989					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Digitale Bildanalyse		Liste IM2	Modulnummer 8130	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Dörner, Schulz, Schwanecke		Verantwortlich Schwanecke
Inhaltliche Voraussetzungen • Lineare Abbildungen, Matrizen, Vektoren, Funktionen mehrerer Veränderlicher, Partielle Ableitungen, Gradienten, Integralrechnung					
Lernziele Die digitale Bildverarbeitung ist in flexiblen Fertigungslinien wesentliche Voraussetzung für eine automatisierte Prozesssteuerung, Prozessanalyse und Qualitätssicherung. Sie wird eingesetzt bei variablen Entscheidungsprozessen (z.B. Biometrie, Verkehrssteuerung und -lenkung). Stark expandierende Einsatzgebiete stellen medizinische Assistenz- und Diagnosesysteme dar. Die Studierenden sollen nach Abschluss des Moduls befähigt sein, • Problemstellungen für die Einsatzmöglichkeit der digitalen Bildverarbeitung zu bewerten, • Problemlösungen für den Praxiseinsatz mithilfe von Standard-Tools zu erstellen, • kreative Lösungen für Problemstellungen zu erarbeiten, die nicht mit Standard-Tools gelöst werden können. Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei.					
Inhalte • Übergang von der realen 3D-Welt auf 2D-Bilder durch orts- und zeitabhängige Quantisierung • Aufbau von CCD-Bildsensoren • Globale Operationen im Ortsbereich • Morphologische Operatoren und Verfahren • Statistische Verfahren und Klassifikationen • lineare, bilineare und trilineare Interpolationen • Bild-Transformation in andere Räume • Bildgebende Verfahren in unterschiedlichen Anwendungsfeldern • Algorithmen zur Analyse von Strukturen (z.B. in medizinischen Bildern) • Stereoskopische Bildverarbeitung • Farbmatrik, Farbbilder, Farbbildsensoren • Echtzeitfähigkeit					
Literatur B. Jähne: Digitale Bildverarbeitung, Springer, 2010 R. C. Gonzalez, R. E. Woods: Digital Image Processing, Prentice Hall International, 2008 A. Nischwitz, M. W. Fischer, P. Haberäcker: Computergrafik und Bildverarbeitung: Alles für Studium und Praxis, Vieweg+Teubner, 2007 J. Steinmüller: Bildanalyse: Von der Bildverarbeitung zur räumlichen Interpretation von Bildern, Springer, 2008					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien)					

Modulbezeichnung / Prüfungsfach Human Computer Interaction		Liste IM2	Modulnummer 8140	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Dörner, Schulz, Schwanecke		Verantwortlich Dörner
Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse in graphischer Datenverarbeitung • Grundkenntnisse in der Realisierung von User Interfaces					
Lernziele Die Studierenden sind nach Abschluss der Veranstaltung in die Lage versetzt, • wissenschaftliche Erkenntnisse im Bereich der Human-Computer-Interaction (HCI) auf die Realisierung von Benutzungsschnittstellen in individuellen Anwendungskontexten anzuwenden • qualitativ hochwertige Designs von Benutzungsschnittstellen zu konzipieren • vertiefte Fertigkeiten in der softwaretechnischen Umsetzung von HCI in Softwaresystemen entwickelt zu haben • auf Erfahrungen in den Besonderheiten bei der Durchführung von anspruchsvollen Projekten im Bereich HCI zurückgreifen zu können • eigene Ideen und Innovationen mit wissenschaftlicher Methodik untersuchen und bewerten zu können Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen sowie zur Erweiterung von spezifischen technologischen Kompetenzen bei.					
Inhalte • Grundlegende Paradigmen, Modelle und Theorien im Bereich Interaktion und Kommunikation • Gestaltungsraum für HCI – ausgewählte Beispiele • Design Prozesse und Gestaltungsrichtlinien • Softwaretechnische Umsetzung von HCI • Hardwaregrundlagen • Innovative computergraphische Techniken und Methodologien für die Realisierung von HCI • Wissenschaftliche Methodik im Bereich HCI					
Literatur B. Shneiderman et.al.: Designing the User Interface: Strategies for Effective Human-Computer Interaction (5th Ed.), Addison-Wesley, 2009 P. Cairns, A.L. Cox (Eds.): Research Methods for Human-Computer Interaction, Cambridge University Press, 2008 Ausgewählte Originalliteratur					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien)					

Modulbezeichnung / Prüfungsfach Industrial and Home Automation		Liste IM2	Modulnummer 8150	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Kaiser, Werntges		Verantwortlich Kaiser
Inhaltliche Voraussetzungen • Grundlagen Rechnernetze, Softwaretechnik, Betriebssysteme • Mathematik: Komplexe Zahlen, Grundlagen zur Fourier-Transformation, Differentialgleichungen • Grundbegriffe der Systemtheorie					
Lernziele Nach erfolgreichem Abschluss dieses Moduls sollten die Studierenden • die grundlegenden Prinzipien und Wirkungsweisen eines Reglers verstehen und anwenden können • mit der Funktionsweise und den besonderen Eigenschaften von Netzen (Feldbussen, Sensornetzen) in Automatisierungsanwendungen vertraut sein • (Muster-)Beispiele von Implementierungen z.B. Klimatechnik, Hochregallager, Gebäudesystemtechnik gesehen haben • Gebäudeautomationstechnik am Beispiel KNX (ISO/IEC 14543) verstehen, planen und exemplarisch in Betrieb nehmen können • die Methoden der Validierung von Automatisierungssoftware und daraus resultierende Konsequenzen kennen • fachübergreifendes Verständnis z.B. für Fertigungs- und Logistik-Prozesse entwickelt haben • die Auswirkung von Automatisierungslösungen unter ökonomischen und gesellschaftlichen Aspekten erkennen können • den Aufwand von Automatisierungslösungen beurteilen können Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen, zur Erweiterung von spezifischen technologischen Kompetenzen und zu fachübergreifenden technischen Systemkenntnissen bei.					
Inhalte • Ziele und Motivation von/für Automatisierungslösungen • Grundlagen der Regelungstechnik • Regelalgorithmen, Prinzipien der digitalen Regelung • Signalaufbereitung, -filterung • Automatisierungs-Hardware (SPS, PLS, Netzadapter) • SCADA-Software • Schnittstellen zwischen (Automatisierungs-)Rechnern und Umwelt • Horizontale Integration, Vernetzung von Automatisierungs-Komponenten (Feldbus-, Sensornetze, Powerbus, etc.) • Vertikale Integration, Integrationstechniken auf IP-Basis, Anbindung an administrative Systeme (z.B. SAP) • Validierung von Automatisierungssystemen • Lifecycle und Kosten von Automatisierungslösungen • Drahtlose, energie-autarke eingebettete Systeme in der Gebäudesystemtechnik • Beispiele / Anwendungen von Automatisierungslösungen in der Gebäudeleittechnik, Fertigungs und Prozesstechnik und im Bereich Automotive.					
Literatur U.Rembold/P.Levi: Realzeitsysteme zur Prozessautomatisierung, Hanser Verlag, 1994 Jan Lunze: Automatisierungstechnik. 2. Auflage. Oldenbourg Wissenschaftsverlag, 2007 K.W. Früh, U. Maier: Handbuch der Prozeaaaautomatisierung. 4. Auflage. Oldenbourg Industrieverlag, 2009 KNX specifications (EN 50090) V 2.0, KNX Association (www.knx.org), 2009 Dietrich/Kastner/Sauter (Hrsg.): EIB Gebäudebussystem, Hüthig, 2000					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien) • prakt. Übungen, z.T. mit programmierbaren Demo-Geräten • ggfs Exkursionen zu interessanten Projekten (Kläranlage, HRL, Pharmafertigung)					

Modulbezeichnung / Prüfungsfach Interaktive 3D-Computergrafik		Liste IM2	Modulnummer 8160	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Dörner, Schulz, Schwanecke		Verantwortlich Dörner
Inhaltliche Voraussetzungen • Objektorientierte Softwareentwicklung: Programmierkenntnisse • Computergrafik: Grundkenntnisse der generativen Computergrafik und über GDV-Systeme					
Lernziele Die Studierenden sollen nach Abschluss der Veranstaltung erworben haben: • Verständnis des interaktiven Modellierungsprozesses sowie der Interaktion von 3D-Objekten (z.B. Kollisionserkennung) • Fertigkeiten in der Erstellung von graphisch-interaktiven Anwendungen, die Echtzeitbedingungen einhalten • Verständnis spezieller Renderingtechniken (Volume Rendering, Non-photorealistic Rendering, Image-Based Rendering, Radiosity, Raytracing ...) • Tiefere praktische Erfahrungen in einer Grafik API (z.B. OpenGL Extensions, DirectX, ...) sowie der Grafikkartenprogrammierung (Vertex-, Pixelshader, CG, ...) Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei.					
Inhalte • Modellierung auf Netzen (Netz-Erzeugung, Optimierung, Dezimierung, Komprimierung, Parametrisierungen, Hierarchische Darstellungen) • Kollisionserkennung • Erweiterte Funktionalität von Grafik APIs (OpenGL Extensions, ...) • Volume Rendering • Spezielle Rendering-Techniken (Non-photorealistic Rendering, Image-Based Rendering, Light fields, Komprimierung, Lumigraph, ...) • Grafikkartenprogrammierung (Vertex-, Pixel-Shader, "Hochsprachen" wie z.B. GLSL) • Globale Beleuchtungsverfahren (Rendering-Gleichung, Radiosity, Monte-Carlo-Methoden) • Interaktion in der 3D Computergraphik					
Literatur Shirley et.al.: Fundamentals of Computer Graphics, AK Peters, 2006 Angel: Interactive Computer Graphics, 5th Ed. Addison-Wesley, 2008 Akenine-Möller et.al.: Real-time Rendering, 3rd Ed., AK Peters, 2008					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien)					

Modulbezeichnung / Prüfungsfach Internet der Dinge		Liste IM2	Modulnummer 8170	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Gergeleit, Kröger		Verantwortlich Gergeleit
Inhaltliche Voraussetzungen • Grundkenntnisse über eingebettete Systeme • Grundkenntnisse über die Vernetzung von Rechnern • Grundlagen der Verteilten Systeme					
Lernziele Die elektronische Vernetzung von Alltagsgegenständen durch z.B. intelligente netzwerkfähige Etiketten mit eigener Sensorik und Aktorik wird Produktion, Logistik und Qualitätsmanagement von Gütern aber auch die Abläufe in privaten Umgebungen stark beeinflussen. Nach Abschluss des Moduls haben die Studierenden erworben: • Vertiefung des Fachwissens im Spezialisierungsgebiet Vernetzte Systeme durch Kenntnis der Grundlagen, technologischen Prinzipien, Einsatzgebiete und Grenzen des Internets der Dinge • Vertiefung des Urteilsvermögens über neue technologischen Entwicklungen auf diesem Gebiet • Entwicklung fachübergreifender Kompetenzen durch Integration z.B. mit Fertigungs- und Logistik-Prozessen sowie durch Analyse und Beurteilung von Auswirkungen auf betriebliche Prozesse, ökonomische und gesellschaftliche Erfordernisse und Randbedingungen • Fertigkeiten in der Beherrschung der entsprechenden Technologien durch Teilnahme am Praktikum (wie z.B. RFID-Systeme und Sensornetze) soweit, dass diese in zukünftigen größeren Projektkontexten eingesetzt werden können Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei.					
Inhalte • Internet der Dinge: Grundlegende Definitionen und Abgrenzungen • Hardware-Grundlagen, insb. RFID-Technologien • Netzwerke für das Internet der Dinge, insb. Feldbusse und Low-Power-Funktechnologien • Ad-hoc-Netze und Sensornetze • Multi-Sensor Data Fusion • Selbstlokalisierung und Tracking • Middleware für das Internet der Dinge • Integration mit dem klassischen Internet • Betriebswirtschaftliche Aspekte des Internet der Dinge • Anwendungsgebiete: insb. Logistik, industrielle und Heim-Automatisierung, Medizintechnik • Durchführung eines Praktikums					
Literatur Fleischer, Mattern (Herausgeber): Das Internet der Dinge: Ubiquitous Computing und RFID in der Praxis. Visionen, Technologien, Anwendungen, Handlungsanleitungen, Springer, 2005 Zitterbart, Baumung: Basissoftware für drahtlose Ad-hoc- und Sensornetze, Kit Scientific Publishing, 2009					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Aufgabenstellungen als PDF					

Modulbezeichnung / Prüfungsfach Komplexitätstheorie		Liste IM2	Modulnummer 8180	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Barth, Geib, Reith		Verantwortlich Reith
Inhaltliche Voraussetzungen • einfache Grundbegriffe der elementaren Logik • Umgang mit den wichtigsten Beweisverfahren • sichere Beherrschung des Induktionsprinzips und der vollständigen Induktion					
Lernziele Nach Abschluss des Moduls • können die Studierenden selbstständig komplexe Beweistechniken auf Probleme der Berechenbarkeits- und Komplexitätstheorie anwenden • kennen die Studierenden typische unentscheidbare Probleme • kennen die Studierenden die üblichen Komplexitätsklassen, typische vollständige Probleme und ihre Bedeutung in der Praxis • sind die Studierenden in der Lage, ihnen unbekannte NP-vollständige Probleme zu erkennen, und kennen Methoden, mit diesen in der Praxis umzugehen. Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von formalen, mathematischen und algorithmischen Kompetenzen bei.					
Inhalte • Raum- und Zeitkomplexität • Beziehungen zwischen den Komplexitätsklassen • Die Hierarchiesätze • Die Klasse P • Die Klasse NP • NP-Vollständigkeit • Der Satz von Cook • Weitere NP-vollständige Probleme • Approximierbarkeit (TSP, Partitionierung) • Die Polynomialzeithierarchie • Randomisierte Komplexitätsklassen • Raumbeschränkte Berechnungen • Komplexität paralleler Berechnungen					
Literatur Michael Sipser, Introduction to the Theory of Computation, Thompson, 2006 Sanjeev Arora, Boaz Barak, Computational Complexity - A Modern Approach, 2009 Uwe Schöning, Theoretische Informatik - kurzgefasst, Spektrum Verlag, 2008					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Kryptologie		Liste IM2	Modulnummer 8190	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Geib, Reith		Verantwortlich Geib
Inhaltliche Voraussetzungen • Grundverständnis über die wichtigsten Sicherheitsbegriffe (Sicherheitsmodell, Sicherheitsmechanismus, Sicherheitsarchitektur etc.) • Einschätzung über Gefahren, Angriffe, Schwachstellen und Risiken im Zeitalter der Vernetzung und des E-Commerce • Kenntnisse über die Grundbestandteile eines Sicherheitskonzeptes und über den Ablauf eines Sicherheitsprozesses					
Lernziele Wer in einem Unternehmen eine sichere Informations- und Kommunikationsinfrastruktur etablieren möchte, kommt an der Anwendung moderner Kryptographie (Kryptosysteme) nicht vorbei. Dies gilt vor allem, wenn über das Unternehmensnetz sensible Geschäftsprozesse abgewickelt werden oder privater Datenverkehr fließt. Die Schwerpunkte der Lehrveranstaltung liegen daher auf den kryptographischen Verfahren zur Einrichtung einer sicheren Kommunikation über mehr oder weniger bedrohte Netze. Nach Abschluss der Lehrveranstaltung sind die Studierenden befähigt, • Verfahren zur Authentifizierung von Teilnehmern verstehen und auswählen zu können, • Methoden der Informationsverschlüsselung einordnen und analysieren zu können, • Vorkehrungen zur Datenintegrität beurteilen und anwenden zu können und • Mechanismen zum Schlüsselmittelaustausch einsetzen zu können. Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von formalen, mathematischen und algorithmischen Kompetenzen bei und erweitern die Methodenkompetenzen und die Analyse-, Design- und Realisierungskompetenzen.					
Inhalte • Grundlagen der Rechnerarithmetik (Addition, Multiplikation, Inversion, Exponentiation) • Symmetrische Kryptosysteme (Blockchiffren, AES, Stream Ciphers) • Public-Key Kryptographie (RSA-Kryptosystem, Diffie-Hellman-Schlüsselaustausch) • Hashfunktionen und Digitale Signaturen (MD4, MD5, Blockchiffren-basierte Hashfunktionen) • Zero-Knowledge Protokolle (Fiat-Shamir Protokoll) • Elliptic Curve Cryptography (ElGamal Signaturen und DH Key-Exchange)					
Literatur Albrecht Beutelspacher: Kryptographie. Vieweg, Braunschweig, 2009 Johannes Buchmann: Einführung in die Kryptographie. Springer-Verlag, Berlin, 2008 David Kahn: The Codebreakers. Macmillan, New York, 1967 Ditmar Wütjen: Kryptographie - Grundlagen, Algorithmen, Protokolle. Spektrum Akademischer Verlag, 2008					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript, Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Management verteilter Systeme und Anwendungen		Liste IM2	Modulnummer 8200	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Kröger, Schmid		Verantwortlich Kröger
Inhaltliche Voraussetzungen • Verteilte Systeme: Grundlagen • Rechnernetze: Grundlagen					
Lernziele Der Betrieb komplexer verteilter IT-Systeme und unternehmenskritischer Anwendungen unter geforderten Qualitätsgütemerkmalen ist eine schwierige Aufgabe mit besonders hoher Bedeutung für die Praxis. Nach Abschluss des Moduls sollen die Studierenden erzielt haben: <ul style="list-style-type: none"> • Vertiefung des Fachwissens und der technologischen Kompetenzen bis zum Stand der Forschung auf dem Gebiet des IT-Managements • Vertiefung der Analyse- und Design-Kompetenz für Problemstellungen des Managements verteilter Systeme und Anwendungen einschl. Nutzung geeigneter mathematischer und von Informatik-Modellen • Entwicklung eines ausgeprägten Urteilsvermögens für methodische und praktische Aspekte des Managements verteilter Systeme und Anwendungen • Vertiefung fachübergreifender Kompetenzen durch Betrachtung der Wechselwirkungen mit Geschäftsprozessen und betriebswirtschaftlichen Aspekten wie z.B. Accounting • Entwicklung von Fertigkeiten in der Beherrschung von Management-Technologien durch das Praktikum Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei.					
Inhalte <ul style="list-style-type: none"> • Architekturen vernetzter Systeme und Anwendungen • Grundlagen des IT-Managements • Managementarchitekturen und ihre Sichten, verbreitete Managementarchitekturen • Standardisierte Vorgehensweisen für das Enterprise Management (z.B. ITIL, ISO) • Managementwerkzeuge und -plattformen • IT Service Management • Überblick über Methoden zur Leistungsbewertung • Instrumentierung von Systemen und Anwendungen • Automatisierung von IT-Management-Prozessen • Service Level Management • Selbstmanagement (Self-X) • Entwicklungswerkzeuge • Ausgewählte Beispiele und Lösungen 					
Literatur Hegering, Abeck, Neumair: Integriertes Management vernetzter Systeme, dpunkt-Verlag, 1999 Beims: IT-Service Management in der Praxis mit ITIL3: Zielfindung, Methoden, Realisierung, Hanser, 2009 Keller: IT-Unternehmensarchitektur: Von der Geschäftsstrategie zur optimalen IT-Unterstützung, dpunkt, 2007 Jain: The Art of Computer Systems Performance Analysis, Wiley, 1991					
Medienform / Unterrichtssprache <ul style="list-style-type: none"> • Veranstaltungsspezifische Web-Seite • Projektaufgabe in schriftlicher Form 					

Modulbezeichnung / Prüfungsfach Middleware-Architekturen		Liste IM2	Modulnummer 8210	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Kröger, Weitz		Verantwortlich Weitz
Inhaltliche Voraussetzungen • Grundlagen Rechnernetze, Interprozesskommunikation, Netzwerkprogrammierung • Komponentenbegriff der Softwaretechnik, Schnittstellenentwurf • Grundlagen Datenbanken, Transaktionsbegriff					
Lernziele Nach Abschluss des Moduls können die Studierenden • für eine konkrete Problemstellung geeignete Middleware-Lösungen auswählen, adaptieren und einsetzen, • Erweiterungen für Middleware-Lösungen selbst entwerfen und realisieren, wobei Kenntnisse aktueller Algorithmen und Protokolle genutzt werden, • die Wiederverwendbarkeit einer Lösung durch die Wahl passender Schnittstellen sicherstellen. Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei.					
Inhalte • Middleware-Begriff, Abgrenzung, Anforderungen (z.B. Verlässlichkeit, Skalierbarkeit, Transparenz, Sicherheit) • Middleware-Kategorien und ihre zugrundeliegenden Algorithmen und Protokolle, Beispiele für Realisierungen • Transaktionale Middleware • Nachrichtenorientierte Middleware (MOM) • Event-orientierte Middleware • Objektorientierte und komponentenorientierte Middleware • Dienstorientierte Middleware • Dokumenten-orientierte Middleware • Middleware für spezielle Anwendungsbereiche (Peer-to-Peer-Systeme, Mobile Anwendungen, Media Streaming, E-Commerce, Datenzentrierte Anwendungen)					
Literatur Tanenbaum, van Steen: Verteilte Systeme: Prinzipien und Paradigmen, Pearson, 2007 Heinzl et al: Middleware in Java, Teubner, 2005 Völter et al: Remoting Patterns: Foundations of Enterprise, Internet and Realtime Distributed Object Middleware, Wiley, 2004					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Folien, Vorlesungsunterlagen und Übungsblätter					

Modulbezeichnung / Prüfungsfach Non-Standard Datenbanken		Liste IM2	Modulnummer 8220	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Heimrich		Verantwortlich Heimrich
Inhaltliche Voraussetzungen • Grundlagen Datenbanken					
Lernziele Vermehrt werden Spezialdatenbanken für verschiedenste Anwendungen eingesetzt, die mit ihren Funktionalitäten die entsprechenden Gebiete in den Naturwissenschaften und den Wirtschaftswissenschaften effizient unterstützen. Die Studierenden sind nach erfolgreichem Abschluss der Veranstaltung in der Lage, • Unterscheidungsmerkmale und Einsatzgebiete unterschiedlicher Non-Standard Datenbanken zu kennen und zu bewerten, • Datenmodellierung mit unterschiedlichen Non-Standard Datenbanken umzusetzen, • Entwurf und Realisierung eines Projekts mit mehreren Studierenden durchzuführen. Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei.					
Inhalte • Data Warehouse-Anwendungen (Multidimensionale Datenbanken, Modelliermethoden für MDBs, Online Analytical Processing, Praktische Einsatzgebiete) • Geo-Datenbanken (Struktur und Informationsreferenzierung, Abfrageproblematiken, Anwendungsbeispiele) • Objektorientierte Datenbanken (Theorie und Modellierung, Abfragen in einem OODBMS) • Datenbanken und XML • Web-Datenbanken (Zugriffsmechanismen) • Semantik in Datenbanken					
Literatur Bauer, Günzel: Data-Warehouse-Systeme, dpunkt-Verlag, 2008 Lehner: Datenbanktechnologie für Data-Warehouse-Systeme, dpunkt-Verlag, 2003 Adamson: Venerable, Data Warehouse Design Solutions, John Wiley & Sons, 1998 Cattell: Object Database Standard ODMG 3.0, Morgan Kaufmann Publishers, 2000 Lehner: XQuery, dpunkt-Verlag, 2004 Bhowmick, Madria, Ng: Web Data Management, Springer, 2003 Daconta, Oberst, Smith: The Semantic Web, John Wiley & Sons, 2003					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Software-Qualität		Liste IM2	Modulnummer 8230	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Bearbeitung (Praktikumsaufgaben) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Igler	Verantwortlich Igler	
Inhaltliche Voraussetzungen • Softwaretechnik: SW-Entwicklungsprozess, OO Analyse und Design, Grundlagen Testing					
Lernziele Die Studierenden sind nach Abschluss des Moduls in der Lage, • Qualitätsanforderungen in allen Phasen des Software-Entwicklungsprozesses (und für den Prozess selbst) projektadäquat zu definieren und zu begründen • geeignete Verfahren zur Überprüfung der Einhaltung auszuwählen • die notwendigen Aktivitäten zur organisieren bzw. (ggf. automatisiert) durchzuführen. Die erworbenen Fähigkeiten vertiefen damit die Analyse-, Design- und Realisierungskompetenzen und das entsprechende Methodenwissen.					
Inhalte • Qualitätsmodelle und Qualitätsmanagement • Software-Entwicklungsprozesse und Prozessqualität • Analyse und Beurteilung von Software-Architekturen • Komponentenqualität (Testing funktionaler und nichtfunktionaler Anforderungen, Verifikation, Metriken) • Prüfprozesse und -werkzeuge					
Literatur Liggesmeyer: Software-Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum, 2009 Clements et al: Evaluating Software Architectures, SEI Series in Software Engineering, Addison-Wesley, 2004 Balzert: Lehrbuch der Softwaretechnik: Softwaremanagement, Spektrum, 2008 Spillner et al: Praxiswissen Softwaretest - Testmanagement, Aus- und Weiterbildung zum Certified Tester - Advanced Level nach dem ISTQB-Standard, dpunkt, 2008					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Folien, Vorlesungsunterlagen und Übungsblätter					

Modulbezeichnung / Prüfungsfach Systemarchitektur		Liste IM2	Modulnummer 8240	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Gergeleit, Kröger		Verantwortlich Kröger
Inhaltliche Voraussetzungen • Rechnernetze: Grundlagen • Softwaretechnik: Grundlagen • Verteilte Systeme: Grundlagen					
Lernziele Bei Entwurf und Konstruktion von Hard- und Softwaresystemen, angefangen von kleinen eingebetteten Systemen bis hin zu hochkomplexen Informationssystemen, werden immer wieder die gleichen grundlegenden Architekturprinzipien gewählt, wenn auch oft auf unterschiedlichen Systemebenen und mit anderer Skalierung. Nach erfolgreichem Abschluss dieser Lehrveranstaltung haben die Studierenden erworben: • Vertiefung des Fachwissens und der technologischen Kompetenzen auf dem Gebiet der Systemarchitektur mit Kenntnis von beispielhaften Umsetzungen • Vertiefung der Analyse- und Design-Kompetenz für Problemstellungen gemäß vorgegebener Entwurfsziele, auch nicht-funktionaler wie Fehlertoleranz, Echtzeitfähigkeit und Sicherheit • Stärkung des Urteilsvermögens für methodische und praktische Aspekte von Systemarchitekturen Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei.					
Inhalte • Einführung in den strukturierten Systementwurf • Grundlegende Konzepte • Hardware-Systemstrukturen • Formen und Alternativen der Systemmodellierung • Kommunikations- und Synchronisationsparadigmen • Konsistenzmodelle • Middleware-Architekturen • Ressourcen-Scheduling • Fehlertoleranzmaßnahmen • Sicherheitsarchitekturen • Leistungsoptimierung • Beispiele von strukturierten Systemen • Beispielhafte Plattformrealisierungen auf verschiedenen Systemebenen					
Literatur Verissimo, Rodrigues: Distributed Systems for System Architects, Kluwer Academic Publishers, 2001 Tanenbaum: Structured Computer Organization. 5. ed., Prentice-Hall, 2006 Tanenbaum, van Steen: Verteilte Systeme, Pearson, 2. Auflage, 2007 Coulouris, Dollimore, Kindberg, Blair: Distributed Systems. Concepts and Design, Pearson, 5. Auflage, 2011 Vogel et al: Software-Architektur, Grundlagen-Konzepte-Praxis, Spektrum-Verlag, 2005					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Selected Topics of Computer Science		Liste IM2	Modulnummer 8250	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Dozenten des Studienbereichs, Gastprofessoren, Lehrbeauftragte		Verantwortlich Schwanecke
Inhaltliche Voraussetzungen (keine)					
Lernziele Successful completion of the module results in <ul style="list-style-type: none"> • detailed knowledge in selected topics of computer science • improved analytical skills for dealing with complex systems • improved formal, mathematical and algorithmic skills by applying formal models related to the given topics • improved judging skills by comparing different designs and implementations in exercises and practicals 					
Inhalte <ul style="list-style-type: none"> • Current topics in computer science at Master level 					
Literatur dependant on selected topics					
Medienform / Unterrichtssprache <ul style="list-style-type: none"> • dependant on selected topics • Language: English 					

Modulbezeichnung / Prüfungsfach Wissensbasierte Systeme		Liste IM2	Modulnummer 8260	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Barth, Krechel, Panitz		Verantwortlich Krechel
Inhaltliche Voraussetzungen • Aussagenlogik: Syntak, Semantik, Resolution					
Lernziele Wissensbasierte Systeme werden zunehmend als gekapselte Komponenten in Anwendungen oder Hardware eingebettet. Die Studierenden haben nach Abschluss des Moduls folgende Fähigkeiten: • sie kennen die Merkmale von Expertensystemen, können diese beurteilen und darauf basierende Lösungsansätze entwickeln, • sie kennen die Theorie zu interner Struktur und Organisation wissensbasierter Systeme und können diese für konkrete Problemlösungen anwenden • sie können Entwurfstechniken zur Entwicklung von wissensbasierten Systemen einsetzen und damit praktisch verwertbare Lösungen entwickeln. Die erworbenen Fähigkeiten tragen damit in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen und zur Erweiterung von spezifischen technologischen Kompetenzen bei.					
Inhalte • Einführung in wissensbasierte Systeme • Struktur wissensbasierter Systeme • Logik und Inferenzmechanismen • Regelbasiertes Reasoning • Blackboard-Architektur • Uncertainty Management • Suche als Problemlösungsmechanismus • Fortgeschrittene Reasoning-Techniken • Wissensakquisition					
Literatur Jackson: Introduction To Expert Systems, Addison Wesley, 1999 Darlington: The Essence of Expert Systems, Prentice Hall, 1999 Gonzales, Dankel: The Engineering of Knowledge-Based Systems, Theory and Practice, Prentice Hall, 2000 Winston: Artificial Intelligence, Addison Wesley, 1993 David, Krivine, Simmons: Second Generation Expert Systems, Springer, 1993 Russel, Norvig: Künstliche Intelligenz: Ein moderner Ansatz, Pearson Studium, 2004 Ausgewählte Originalliteratur					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Robotik		Liste IM2	Modulnummer 8270	Sem. 1, 2	Studiengang Informatik (M.Sc.)
Credits / SWS 5 cp, 150 h 4 SWS	Lehrformen • Vorlesung (2 SWS) • Praktikum (2 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 60 h Anwesenheit (Vorlesung und Praktikum) • 30 h Vor- und Nachbereitung (Vorlesung) • 60 h Bearbeitung (Praktikumsaufgaben)			Dozenten Richter		Verantwortlich Kröger
Inhaltliche Voraussetzungen • Mathematik : Homogene Koordinaten, 3D-Rotationsmatrizen, Matrizenrechnung • Informatik : Der Einsatz und die Anwendung von Mathematikprogrammen wie Mathcad, Matlab, Mathematica, Autodesk Inventor o.ä. • Für Projektarbeiten : Grundkenntnisse in Digitaler Bildverarbeitung und/oder Graphischer Datenverarbeitung und/oder Virtuelle Realitäten und/oder Augmentierte Realitäten					
Lernziele Rationalisierungsmaßnahmen in flexiblen Fertigungslinien für kleine und mittlere Serien, die Laborautomation für Kleinserien-Analysen, die Entwicklung von medizinischen Assistenzsystemen und ein zukünftiger Einsatz von Robotern im Bereich des AAL verlangen einen nicht-standardisierten und häufig sensorgesteuerten Einsatz von mehrachsigen Robotern. Dazu sind Kenntnisse der Bewegungsdynamik, die über die ingenieurmäßige Anwendung von Industrierobotern deutlich hinausgehen, notwendig. Die Studierenden werden befähigt, • Entscheidungskriterien für den Einsatz unterschiedlicher Robotertypen zu entwickeln, • Position und Orientierung von Armteilen mathematisch zu modellieren und zu berechnen, • Gelenkstellungen zu berechnen, • Mehrdeutigkeiten mithilfe von Konstruktionsmerkern aufzulösen, • Simulations-SW anzuwenden, • Nicht-repetitive Bewegungsabläufe mithilfe von Sensoren, z. B. Kameras, zu steuern, • Arbeitsräume zu definieren und zu berechnen, • Bahnplanungen durchzuführen, • Kollisionsräume zu modellieren, • Bewegungsabläufe zu visualisieren.					
Inhalte • Gegenwärtiger Entwicklungsstand der Robotik und deren allgemeine Einsatzmöglichkeiten, • Wirtschaftliche Bedeutung der Robotik in der Fertigungstechnik, • Robotertypen, Gelenkarten, Bewegungsarten, • Sensoren für die Robotik (visuelle, elektronische, mechanische), • Mathematische Verfahren zur Bewegungsberechnung • Denavit-Hartenberg-Notation, • Vorwärtskinematik, • Inverse Kinematik, • Visualisierung von Bewegungsabläufen unter Sensorkontrolle					
Literatur Rieseler, H.: Roboterkinematik - Grundlagen, Invertierung und symbolische Berechnung A Campo, M.: Kollisionsvermeidung in einem Roboter-Simulationssystem					
Medienform / Unterrichtssprache • Powerpoint-Präsentation, Erläuterungen an der Tafel, • Referate der Studierenden, • Virtuelle Darstellung von Bewegungsabläufen.					

Modulbezeichnung / Prüfungsfach 3D-Bildanalyse und -Synthese		Liste VM2	Modulnummer 8510	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Dörner, Schulz, Schwanecke		Verantwortlich Dörner
Inhaltliche Voraussetzungen • Computergrafik: Grundkenntnisse in graphischer Datenverarbeitung • Objektorientierte Softwareentwicklung: Programmierkenntnisse					
Lernziele Die Studierenden sind nach Abschluss des Moduls in der Lage, • geeignete Verfahren und geeignete Hardware für 3D-Bildanalyse und –synthese für ein gegebenes Problem auszuwählen, • diese kritisch zu beurteilen, • Grundverfahren problemorientiert anzupassen, • verfügbare Werkzeuge zu nutzen und • neue Verfahren und dedizierte Software zu entwickeln. Neben der Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen haben die Studierenden ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen vertieft sowie ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams erweitert. Darüber hinaus sollten die Studierenden grundlegendes Wissen über das Zusammenspiel von 3D-Bildanalyse einerseits und 3D-Bildsynthese andererseits erworben haben sowie damit zu realisierende Methodologien (wie z.B. im Bereich Mixed Reality oder im Bereich bildbasiertes 3D-Rendering) sowie Anwendungen (wie z.B. im Bereich Medizin oder allgemein im Bereich Human-Computer-Interaction) kennen und beurteilen können.					
Inhalte • Sensorik (CCD, Farbbildsensoren, ...) • Algorithmen und Verfahren der 3D Bildanalyse (Merkmalsextraktion, Merkmalsklassifikation, statistische Verfahren, ...) • Algorithmen und Verfahren der 3D Bildsynthese (Image Based Rendering, Image Compositing, wahrnehmungsbasierte Verfahren) • Kalibrierung, Registrierung und Tracking • Bilderkennen und Bildverstehen • Kombination von Bildanalyse und Bildsynthese (z.B. Mixed Reality) • Architektur von Systemen der 3D Bildanalyse und -Synthese • Dedizierte Software und Werkzeuge • Echtzeitaspekte • Anwendungen					
Literatur Jähne: Digitale Bildverarbeitung, Grundlagen und Anwendungen, 6. Auflage, Springer-Verlag, 2005 Nixon, Aguado: Feature Extraction and Image Processing, 2. Auflage, Academic Press, 2007 Ausgewählte Originalliteratur					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien)					

Modulbezeichnung / Prüfungsfach Advanced Topics of Computer Science		Liste VM2	Modulnummer 8520	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Dozenten des Studienbereichs, Gastprofessoren, Lehrbeauftragte		Verantwortlich Schwanecke
Inhaltliche Voraussetzungen (keine)					
Lernziele After successfully completing the module, students have • developed in-depth knowledge and expertise in selected topics • improved analytical, design, and implementation skills for complex systems • improved formal mathematical and algorithmic skills by developing formal models related to the given problem • improved judging skills by evaluating system designs and implementations • improved soft skills by working in a small team					
Inhalte • Guided exploration of current topics in applied computer science at Master level • Analysis, design, implementation and evaluation of a prototype in a small group • Presentation of project results					
Literatur dependant on selected topics					
Medienform / Unterrichtssprache • dependant on selected topics • Language: English					

Modulbezeichnung / Prüfungsfach Anwendungsintegration		Liste VM2	Modulnummer 8530	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 150 h Praktische Tätigkeit (Projekt) • 60 h Vor- und Nachbereitung (Theorieteil) • 90 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Barth, Krechel, Weitz, Werntges		Verantwortlich Werntges
Inhaltliche Voraussetzungen • Datenbanken (DB-Entwurf, SQL) • Networking (Grundkonzepte, TCP/IP, socket-API, verbreitete Anwendungsschichtprotokolle wie FTP, http, SMTP) • HTML, XML (Grundkonzepte, XSL, XML-Verarbeitung in einer Programmiersprache (insb. Parsing, Transformation)) • Imperative bzw. OO-Programmierung • Serverseitige Webtechnologien (z.B. Servlets, CGI) • Software-Engineering (Software-Lifecycle, Anforderungsanalyse, Entwurf, Architekturen, Ablauf-Modellierung, Re-Engineering)					
Lernziele Im kommerziellen Umfeld ist die Unterstützung vieler Geschäftsprozesse nur durch Integration verschiedener Teilanwendungen (z.B. von Altsystemen) möglich. Die daraus resultierende Verteilung der Anwendungslogik auf mehrere Rechensysteme wirft dabei eine Reihe von Problemen auf. Nach Abschluss des Moduls • kennen die Studierenden die organisatorischen Voraussetzungen für Anwendungsintegration und können an ihrer Schaffung mitwirken, • können sie einen praxisrelevanten Geschäftsprozess fachlich durchdringen, • können sie spezielle Anforderungen verteilter Anwendungssysteme identifizieren, systematisch erfassen und Risiken abschätzen, • können sie Modellierungs- und Analyseaktivitäten umsetzen, • können sie zur Erstellung verteilter Anwendungen vorhandene Technologien unter Berücksichtigung des Integrations-Aspekts bewerten und auswählen, • können sie auf Basis des Praktikums ein verteiltes Anwendungssystem unter Integration bestehender Fremdsystem-Schnittstellen entwerfen und realisieren, • haben sie ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen und ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams vertieft.					
Inhalte Auswahl aus folgenden möglichen Themen: • Einführung, organisatorische Grundlagen und Voraussetzungen, Funktionssicht vs. Prozess-Sicht • Beispiele für verteilte Anwendungen • Entwurf verteilter Anwendungssysteme (Kriterien, Architekturen, Bewertung) • Integration von Daten (Transformation, Schemafusion) • Replikation / Synchronisation verteilt gehaltener Datenbestände • Unternehmensübergreifende Integrationstechniken • Modellierung, Analyse und technische Unterstützung betrieblicher Abläufe • Geschäftsprozess-Monitoring, Metadaten, Interface-Repositories, Konfiguration • Verschiedene Formen von Daten- bzw. Ablaufschnittstellen in Bestandssystemen • Probleme bei der Integration von Altsystemen (z.B. Wrapper) • Überblick über Integrations-Technologien; Unterschiede, Einsatzgebiete, Auswahlkriterien • Integrationsmuster, EAI- bzw. Integrations-Frameworks • Sicherheitsaspekte (z.B. "Single Sign On", verteilte Benutzerdaten)					
Literatur Sommerville: Software Engineering, Addison-Wesley, 2004 Cummins: Enterprise Integration, Wiley, 2002 Fowler et al: Patterns of Enterprise Application Architecture, 2002 Hohpe, Woolf: Enterprise Integration Patterns, Addison-Wesley, 2004					
Medienform / Unterrichtssprache • Web-Seite zur Veranstaltung • Zusatzmaterialien, Seminare/Projektdokumentation					

Modulbezeichnung / Prüfungsfach Fortgeschrittene Themengebiete der Informatik		Liste VM2	Modulnummer 8540	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 150 h Praktische Tätigkeit (Projekt) • 60 h Vor- und Nachbereitung (Theorieteil) • 90 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Dozenten des Studienbereichs, Gastprofessoren, Lehrbeauftragte		Verantwortlich Schwanecke
Inhaltliche Voraussetzungen (keine)					
Lernziele Nach Abschluss des Moduls haben die Studierenden <ul style="list-style-type: none"> • vertieftes Wissen und Kompetenzen in den ausgewählten Gebieten entwickelt • ihre Kompetenzen und Fertigkeiten bei Analyse, Design und Implementierung komplexer Systeme erweitert und vertieft • ihre formalen mathematischen und algorithmischen Fähigkeiten durch Entwicklung formaler Modelle für die gegebene Problemdomäne erweitert und vertieft • ihr Urteilsvermögen durch Bewertung von Systementwürfen und -implementierungen gefestigt • ihre soziale Kompetenz durch Gruppenarbeit und Präsentation von Projektergebnissen erweitert 					
Inhalte <ul style="list-style-type: none"> • Angeleitete Auseinandersetzung mit einem aktuellen Thema der Angewandten Informatik auf Master-Niveau • Analyse, Entwurf, Implementierung und Bewertung eines Prototypen im Rahmen einer Projektgruppe • Präsentation von Projektergebnissen 					
Literatur abhängig von den ausgewählten Themengebieten					
Medienform / Unterrichtssprache <ul style="list-style-type: none"> • abhängig von den ausgewählten Themengebieten 					

Modulbezeichnung / Prüfungsfach Informationsverarbeitung im Semantic Web		Liste VM2	Modulnummer 8550	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Heimrich, Krechel, Panitz		Verantwortlich Krechel
Inhaltliche Voraussetzungen • Aussagenlogik: Syntak, Semantik, Resolution • XML Grundlagen, XML Schema, DTD					
Lernziele Ziel des Semantic Web ist es, die Bedeutung von Inhalten im WWW für Computer auswertbar zu machen. Durch standardisierte Wissensmodellierung und Verarbeitungsmechanismen sollen Informationen von Maschinen interpretiert und verarbeitet werden. Die Studierenden sind nach Abschluss des Moduls in die Lage versetzt, • die W3C Standards des Semantic Web zu kennen und sinnvoll einsetzen zu können, • komplexe Wissenszusammenhänge modellieren zu können, • sinnvolle Anwendungsgebiete für automatische Inferenzen zu identifizieren. Darüber hinaus kennen die Studierenden an Hand von praktischen Anwendungsgebieten, wie zum Beispiel dem Wissensmanagement mit semantischen Wikis, die Praxisrelevanz der Technologien. Ferner haben die Studierenden vertieft: • ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen, • ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams					
Inhalte • Idee des Semantic Web • RDF, OWL • Logik und Inferenz im Semantic Web • Beschreibungslogik • Regelsprachen • Anwendungen semantischer Technologien • Entwurf und Pflege von Ontologien • Semantische Wikis					
Literatur Breitman, Casanowa, Truszkowski: Semantic Web Concepts, Technologies and Applications, Springer, 2007 Antoniou, van Harmelen: A Semantic Web Primer, MIT Press, 2008 Hitzler, Sure: Semantic Web. Grundlagen, Springer, 2007. Blumauer, Pellegrini: Social Semantic Web Web 2.0 - Was nun?, Springer, 2009 Ausgewählte Originalliteratur					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien)					

Modulbezeichnung / Prüfungsfach Informationsvisualisierung		Liste VM2	Modulnummer 8560	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Dörner, Schulz, Schwanecke		Verantwortlich Dörner
Inhaltliche Voraussetzungen • Grundkenntnisse in graphischer Datenverarbeitung					
Lernziele Informationsvisualisierung dient der Analyse und Interpretation von Daten sowie der zwischenmenschlichen Kommunikation von Information - Informationsvisualisierung ist eine grundlegende Methodologie in der Informations- und Wissensgesellschaft. Die Studierenden sind nach Abschluss der Veranstaltung in die Lage versetzt, • geeignete Visualisierungsverfahren und -systeme für ein gegebenes Problem auswählen zu können, • diese kritisch zu beurteilen, • Grundverfahren problemorientiert anzupassen und • neue Verfahren und dedizierte Software zu entwickeln. Darüber hinaus haben die Studierenden grundlegendes Wissen über Informationsvisualisierungen erworben und können so grundlegende Methodologien in der Informations- und Wissensgesellschaft in einem theoretischen Bezugsrahmen einordnen. Über diese spezifischen Analyse-, Design- und Realisierungskompetenzen hinaus haben die Studierenden • ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen vertieft, • ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams verfestigt					
Inhalte • Terminologie, Ziele der Informationsvisualisierung, Visualisierungspipeline • Das visuelle System des Menschen • Grundlagen der Wahrnehmungspsychologie für Visualisierung (z.B. pre-attentive Wahrnehmung, Gestalt-Gesetze) • Einsatz von Visuellen Variablen (z.B. Farbe, Textur, Form) • Grundlegende Visualisierungstechniken • Symbole, Diagramme und Graphen, Benutzung von Bild vs. Wort • Visualisierungstechniken für multivariate Daten • Visualisierung von Kausalbeziehungen • Informationsvisualisierung mit Raum- und Zeitbezug • Interaktive Visualisierung • Visualisierungssysteme • Bewertung von Visualisierungen - Gütekriterien und Testverfahren					
Literatur C. Ware: Information Visualization - Perception for Design (2nd Ed.), Morgan Kaufmann, 2004 R. Spence: Information Visualization - Design for Interaction, Addison Wesley, 2007 S.K. Card et.al.: Readings in Information Visualization, Morgan Kaufmann, 1999 Ausgewählte Originalliteratur					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien)					

Modulbezeichnung / Prüfungsfach Interaktive Informationssysteme		Liste VM2	Modulnummer 8570	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 150 h Praktische Tätigkeit (Projekt) • 60 h Vor- und Nachbereitung (Theorieteil) • 90 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Barth, Krechel, Schwanecke, Steffen, Weitz		Verantwortlich Steffen
Inhaltliche Voraussetzungen • Entwicklung von Webanwendungen • Datenbank-Entwurf und -Implementierung • Objektorientierte Softwareentwicklung • Grundlagen 3D Computergraphik					
Lernziele Rasch selektierbare, schnell erfassbare und individuell konfigurierbare Informationen auf den unterschiedlichsten Medienplattformen entsprechen dem heutigen Wahrnehmungsverhalten. Dabei muss Nutzung und Bedienung der entsprechenden Systeme einfach sein. Dies begründet die grafische Visualisierung von Inhalten und Navigationskonzepten ebenso wie die interdisziplinäre Zusammenarbeit (konzeptionelle, funktionale, gestalterische und technische Kompetenzen) bei der Anwendungsentwicklung. Am Ende des Semesters können die Studierenden • Gestalterische Varianten bilden und Lösungsansätze objektiv beurteilen • Komplexe Inhalte gliedern und zielgruppenspezifisch visualisieren • Modulare Umsetzungskonzepte erarbeiten • Konsistente Realisierungskonzepte erarbeiten • Konzepte und Entwürfe technisch realisieren und funktional überprüfen Ferner haben die Studierenden • ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen vertieft • ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams verfestigt					
Inhalte • Thematisch wechselnde, praxisbezogene (reale oder fiktive) Projekte zur Erstellung elektronischer Medien (z.B. Informationssysteme, Leit- und Orientierungssysteme; technische, medizinische, naturwissenschaftliche oder wirtschaftswissenschaftliche Lehr- bzw. Lernmodule für unterschiedliche Medien) • Interaktive und multimediale Informationsformen • Anforderungsanalyse (Informationsbedarf, Inhalte, technische Rahmenbedingungen) • Usability (User-Involvement, Case Studies, Testing, Auswertung, Anwendung) • Erstellung von Entwurfs- und Umsetzungsvarianten • Visualisierungskonzept und Gestaltungskonstanten • Medien- und zielgruppenspezifische Erstellung, Kombination und Gewichtung von Assets (Text, Bild, Grafik, Animation und Ton) • Strukturierte selbstorganisierte Durchführung eines Projektes • Technische Feinspezifikation • Technische Realisation, Testing und Implementierung • Projektpräsentation					
Literatur Saffer, D.: Designing for interaction, New Riders, 2009 Dix, A.; Finlay, J. E.; Abowd, G. D. & Beale, R.: Human-Computer Interaction, Prentice Hall, 2003					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Machine Learning		Liste VM2	Modulnummer 8580	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Krechel, Schwanecke		Verantwortlich Krechel
Inhaltliche Voraussetzungen • Grundlagen Wahrscheinlichkeitsrechnung und Statistik • Grundlagen der Logik					
Lernziele Maschinelles Lernen beschäftigt sich mit der rechnergestützten Erzeugung von Wissen und Erfahrung. Es werden Systeme betrachtet, die aus in einer Lernphase gelernten Beispielen verallgemeinern und Gesetzmäßigkeiten herleiten können. Die Studierenden sollen nach Abschluss der Lehrveranstaltung in die Lage versetzt sein, • geeignete Verfahren zur Lösung eines gegebenen Problems aus dem Bereich des maschinellen Lernens auswählen zu können, • diese kritisch beurteilen können, • Grundverfahren problemorientiert anpassen können und • neue Verfahren und dedizierte Software entwickeln zu können Darüber hinaus haben die Studierenden grundlegendes Wissen über maschinelles Lernen erworben und Anwendungen im Bereich der visuellen Systeme (z.B. Robotik oder Handschriftenerkennung), Dokumentenanalyse oder Data Mining kennengelernt. Ferner haben die Studierenden • ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen vertieft • ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams trainiert					
Inhalte • Überwachtes und Unüberwachtes Lernen • Konzeptlernen • Genetische Algorithmen • Selbstorganisierende Karten • Neuronale Netze • Automatische Klassifikation (Bayes-Netze, Entscheidungsbäume, Multilayer Perceptron, Support Vector Machines, ...) • Clustering (K-means, mean-shift, SOM, ...) • Analoges Schließen • Induktive Logische Programmierung					
Literatur Bishop: Pattern Recognition and Machine Learning, Springer, 2008 Alpaydin, Linke: Maschinelles Lernen, Oldenburg, 2008 Duda, Har, Stork: Pattern Classification, John Wiley & Sons, 2000 Mitchell: Machine Learning, McGraw-Hill, 1997 Russel, Norvig: Künstliche Intelligenz: Ein moderner Ansatz, Pearson, 2004 Ausgewählte Originalliteratur					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien)					

Modulbezeichnung / Prüfungsfach Mobile Anwendungen		Liste VM2	Modulnummer 8590	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 150 h Praktische Tätigkeit (Projekt) • 60 h Vor- und Nachbereitung (Theorieteil) • 90 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Barth, Gergeleit		Verantwortlich Gergeleit
Inhaltliche Voraussetzungen • Grundkenntnisse über die Vernetzung von Rechnern • Grundlagen der Web-basierten Anwendungen					
Lernziele Durch die allgegenwärtige Vernetzung und die steigende Leistungsfähigkeit mobiler Endgeräte von Smartphones bis zu eingebetteten Internet-fähigen Systemen in anderen technischen Geräten werden Internet- und Web-basierte Anwendungen zunehmend mobil. Dies erfordert sowohl beim Design der Anwendung neue Ansätze als auch ein vertieftes Verständnis der unterliegenden Infrastruktur. Nach Abschluss dieser Lehrveranstaltung sind die Studierenden in die Lage versetzt, • die Begriffe, Konzepte und Techniken von mobilen Anwendungen zu verstehen, • problem- und marktorientiert bestehende mobile Anwendungen bewerten und auswählen zu können, • neue Anwendungsszenarien für mobile Anwendungen zu erkennen und • selbst mobile Anwendungen zu entwerfen und im Rahmen von Anwendungsframeworks zu realisieren und zu betreiben. Neben diesen spezifischen Analyse-, Design- und Realisierungskompetenzen sowie spezifischen technologischen Kompetenzen haben die Studierenden vertieft: • ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen • ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams					
Inhalte • Einführung (Definition und Kategorisierung mobiler Anwendungen, Geschichte) • Mobilfunkmarkt • Geräteklassen • Grundlagen der technologischen Infrastruktur (z.B. WLAN, GPRS, UMTS) • Betriebssysteme für mobile Geräte (z.B. Android, Windows Mobile) • Middleware und Application Frameworks (z.B. J2ME, OSGi) • Service Discovery (Entdeckung und Einbindung, Beschreibung von Syntax und Semantik) • Datensynchronisation, lokale Datenhaltung auf mobilen Geräten, Einsatz von Online/Offline-Lösungen • Personalisierung und Kontextsensitivität von mobilen Anwendungen • Design und Umsetzung von Benutzerschnittstellen für verschiedene mobile Plattformen mit unterschiedlichen Ein-/Ausgabe-Möglichkeiten und Dimensionen • Location Based Services • Mobile Zahlungsverfahren • Sicherheitsaspekte					
Literatur Schiller: Mobilkommunikation, Pearson Studium, 2003 Fuchß: Mobile Computing, Hanser, 2009 Becker, Pant: Android - Grundlagen und Programmierung, dpunkt.verlag, 2009					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Aufgabenstellungen als PDF					

Modulbezeichnung / Prüfungsfach Moderne Verfahren der SW-Entwicklung		Liste VM2	Modulnummer 8600	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Iglter, Weitz		Verantwortlich Weitz
Inhaltliche Voraussetzungen • Softwaretechnik: SW-Entwicklungsprozess, OO Analyse und Design, UML • Objektorientierte Programmierung • Datenbanken: Entwurf, SQL					
Lernziele Nach Abschluss des Moduls • können die Studierenden neuere Methoden und Werkzeuge der Softwareentwicklung einschätzen und bewerten, • sind sie in der Lage, insbesondere die Entwurfs- und Implementierungsphasen größerer Software-Entwicklungsaufgaben systematisch zu analysieren und unter Einsatz fortgeschrittener Methoden und Werkzeuge tragfähige Lösungsansätze zu entwickeln, • sind sie in der Lage, neuere Trends der Softwareentwicklung zu analysieren und deren Anwendbarkeit für konkrete Problemstellungen (auch im Vergleich zu herkömmlichen Konzepten) zu beurteilen, • können sie dies im Rahmen eines größeren, im Team zu bearbeitenden Software-Entwicklungsprozesses demonstrieren. Neben dem Erwerb dieser Analyse-, Design- und Realisierungskompetenzen haben die Studierenden ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen vertieft.					
Inhalte • Fortgeschrittene Software-Entwicklungsprozesse • Modellierung und Metamodellierung • Neuere Entwicklungen bei Programmiersprachen / -plattformen • Aktuelle Konzepte wie Generative Softwareentwicklung, modellgetriebene Softwareentwicklung, domänenspezifische Sprachen • Werkzeugeinsatz • Berücksichtigung spezieller nichtfunktionaler Anforderungen					
Literatur Völter et al: Modellgetriebene Softwareentwicklung: Techniken, Engineering, Management, dpunkt, 2007 Kleppe: Software Language Engineering: Creating Domain-Specific Languages Using Metamodels, Addison-Wesley, 2008					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Vorlesungsunterlagen zum Theorie-Teil					

Modulbezeichnung / Prüfungsfach Multimedia-Kommunikationssysteme		Liste VM2	Modulnummer 8610	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Geib, Gergeleit		Verantwortlich Gergeleit
Inhaltliche Voraussetzungen • Kenntnisse über die Vernetzung von Rechnern, speziell IP-Netze					
Lernziele Durch die Konvergenz der Geräte, der Dienste und der klassischen Telekommunikations- und Datennetze zu den sog. Next-Generation-Networks (NGNs) entstehen neue Anforderungen an die Architekten, Implementierer und Betreiber solcher Infrastrukturen. Nach Abschluss des Moduls • haben die Studierenden ihre Kenntnisse über die technischen Grundlagen der heute üblichen vernetzten Multimedia-Anwendungen, wie z.B. VoIP und Video-Streaming und die Architektur eines All-IP NGNs, vertieft und können diese im entsprechenden Anwendungskontext besser beurteilen, • haben sie die Fähigkeit vertieft, anwendungsspezifische Anforderungen an Multimedia-Kommunikationssysteme zu analysieren und in technische Spezifikationen umzusetzen, • haben sie die Erfahrung durch praktische Projektarbeit erhalten, solche Systeme, bzw. einzelne ihrer Komponenten, selbst zu entwickeln, zu parametrisieren, zu betreiben und zu managen. Neben dem Erwerb dieser spezifischen Analyse-, Design- und Realisierungskompetenzen haben sie ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen und ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams vertieft.					
Inhalte • Dienste und funktionale Anforderungen • Verkehrsmodellierung und Dienstgüte • Architektur und Komponenten eines NGNs • Medienformate für Audio (speziell Sprache) und Video • Protokolle zur Medienübertragung in IP-Netzen (z.B. RTP und RTCP, Multicast) • Protokolle zum Abruf und zur Steuerung von Medienströmen (z.B. RTSP, H.323, SIP) • IPv6 und QoS-Techniken (IntServ, DiffServ, Policy-Based Queueing, IEEE802.1p/Q, MPLS, Layer-3-Switching) • Multimedia Networking Systeme mit Zugangsnetzen (xDSL, drahtlose Zugänge) • Spezifische Sicherheitsaspekte in Multimedia-Netzen • Beispiele für Planung und Betrieb von NGNs • Durchführung praktischer Experimente mit Multimedia Networking Anwendungen mit und ohne QoS.					
Literatur Badach: Voice over IP - Die Technik, Hanser, 2007 Siegmond: Technik der Netze: Neue Ansätze: SIP in IMS und NGN, Hüthig, 2009 Heyna, Briede, Schmidt: Datenformate im Medienbereich, Hanser, 2003					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach Parallele und verteilte Algorithmen		Liste VM2	Modulnummer 8620	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Kröger, Reith		Verantwortlich Kröger
Inhaltliche Voraussetzungen • Verteilte Systeme: Grundlagen • Softwaretechnik: Grundlagen					
Lernziele Entwurf und Implementierung effizienter parallele und verteilte Algorithmen stellen aufgrund der Entwicklungen der Rechnerarchitektur ein wichtiges, zukunftsorientiertes Aufgabengebiet für Informatiker dar. Nach Abschluss der Veranstaltung haben die Studierenden erworben: • Vertiefung der Methodenkompetenz zur Beherrschung methodischer und praktischer Aspekte der Entwicklung paralleler und verteilter Algorithmen • Vertiefung des Fachwissens im gewählten Spezialisierungsgebiet • Vertiefung des Urteilsvermögens durch Bewertung von Ansätzen aus Theorie und Praxis • Vertiefung der Analyse-, Design- und Realisierungskompetenzen im Rahmen der Projektbearbeitung • Vertiefung der kommunikativen Kompetenz durch Erstellung einer wiss. Ausarbeitung und Präsentation von wiss. Ergebnissen und eigenen Projektergebnissen • Vertiefung der sozialen Kompetenz durch Arbeit in kleinen Projektteams					
Inhalte Themengebiete der Vorlesung und des seminaristischen Teils • Maschinenmodelle für parallele und verteilte Verarbeitung (insb. Multicore/Multiprozessor-Systeme, Cluster, Grids, auf Kommunikationsnetzen basierende verteilte Architekturen, GPGPUs, usw.) • Programmierparadigmen für parallele und verteilte Verarbeitung • Abstraktionen für Synchronisation und Kommunikation und deren Programmierschnittstellen in verschiedenen Programmiersprachen • Patterns • Parallele Algorithmen für spezielle Anwendungsgebiete • Implementierungsumgebungen (z.B. Message Passing Interface (MPI), OpenMP, MapReduce/Hadoop, OpenCL) • Grundlagen verteilter Algorithmen • Verteilte Basisalgorithmen (z.B. Wahlalgorithmen, verteilte Terminierung, Schnappschuss, Globale Zeit, Commitment, Versteigerungen) • Spezielle verteilte Algorithmen für bestimmte Anwendungen Durchführung • Vorlesung • Vergabe spezieller Themen zur seminaristischen Aufbereitung • Durchführung eines praktischen Projekts unter Nutzung einer Implementierungsumgebung • Präsentation von Projektergebnissen					
Literatur R. Rauber und G. Rürger: Parallele Programmierung, Springer-Verlag, 2007 G. Bengel et al.: Masterkurs Parallele und Verteilte Systeme, Vieweg+Teubner, 2008 R. Oechsle: Parallele und verteilte Anwendungen in Java, Hanser, 2007 J. JaJa: Introduction to parallel algorithms and architectures, Addison-Wesley, 1992 T. White: Hadoop - The Definitive Guide, O'Reilly, 2009 MPI-Forum: http://www.mpi-forum.org OpenMP: http://openmp.org/wp/					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Projektaufgabe in schriftlicher Form • Ergänzendes eLearning-Material einschl. Simulator für verteilte Algorithmen					

Modulbezeichnung / Prüfungsfach Ubiquitous Computing		Liste VM2	Modulnummer 8630	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Gergeleit, Kröger		Verantwortlich Gergeleit
Inhaltliche Voraussetzungen • Grundkenntnisse über eingebettete Systeme • Grundlagen der Verteilten Systeme					
Lernziele Rechensysteme durchdringen zukünftig das tägliche Leben. Eine vertiefende Auseinandersetzung mit den zugrunde liegenden Klein- und Kleinst-Systemen und ihrer Interaktion mit der Umgebung und Nutzern dient der Zukunftssicherung in technischen Berufsfeldern. Nach Abschluss der Lehrveranstaltung haben die Studierenden erworben: • Vertiefung des Fachwissens im Spezialisierungsgebiet Vernetzte Systeme durch Kenntnis der Grundlagen und technologischen Prinzipien von ubiquitäreren Systemen und Ambient Intelligence • Vertiefung der Analyse- Design- und Realisierungskompetenzen im Rahmen der Projektbearbeitung durch Adaption auf typische Anwendungsfelder • Vertiefung des Urteilsvermögens durch Einordnen und Bewerten entsprechender Systeme von Dritten • Vertiefung der Fertigkeiten, um entsprechende Systeme auf gegebener Basishardware und -software selbst entwerfen und prototypisch umsetzen zu können • Vertiefung der kommunikativen Kompetenz durch Präsentation von eigenen Projektergebnissen • Vertiefung der sozialen Kompetenz durch Arbeit in kleinen Projektteams					
Inhalte • Grundlegende Paradigmen: Interactive Appliances, Ubiquitous Computing, Ambient Intelligence • Kontextmodelle, Kontextakquisition, Kontextspeicherung und Kontextverbreitung • Sensortechnik für die Kontexterfassung • Hardware-Architekturen von Knoten und Interfacing zu Sensoren und Aktoren • Netzwerke und Interaktion in ubiquitären Infrastrukturen • Betriebssysteme für ubiquitäre Infrastrukturen • Basismechanismen des Device- und Service-Discovery • Middleware-Ansätze für ubiquitäre Infrastrukturen • Kollaboration und Selbstorganisation in ubiquitären Infrastrukturen • Kooperationsstrategien, z.B. spieltheoretische und marktbasierende Ansätze • Usability-Aspekte von ubiquitären Systemen • Sicherheit und Privatsphäre in ubiquitären Systemen • Aktuelle Projekte und Anwendungen im Bereich Ubiquitous Computing und Ambient Intelligence • Durchführung eines eigenen Projekts in Kleingruppen					
Literatur Stefan Poslad: Ubiquitous Computing: Smart Devices, Environments and Interactions, John Wiley & Sons, 2009 John Krumm (Herausgeber): Ubiquitous Computing Fundamentals, Crc Pr Inc., 2009					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Aufgabenstellungen als PDF					

Modulbezeichnung / Prüfungsfach Verlässliche Systeme		Liste VM2	Modulnummer 8640	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Geib, Iglar		Verantwortlich Geib
Inhaltliche Voraussetzungen • Grundverständnis über den Aufbau fehlertoleranter Rechensysteme (Fehlerursachen und Fehlerauswirkung, Anforderungen und Zielsetzung, Kritikalität) • Kenntnisse über einfache Beschreibungsmittel in der Zuverlässigkeitstechnik (Zuverlässigkeitsblockdiagramme und Fehlerbäume) • Numerische Verfahren und Analysisgrundlagen (Differentialrechnung, numerische Integration, Fehlerabschätzung) • Statistik und Wahrscheinlichkeitsrechnung (Zufallsvariable, Zufallsprozess, Verteilungsfunktion, Erwartungswert)					
Lernziele Nach erfolgreichem Abschluss des Moduls • überblicken die Studierenden das Thema Funktionssicherheit und Ausfallsicherheit, • kennen sie Schutzmaßnahmen, die ein System weniger fehleranfällig machen gegenüber äußeren Einflüssen sowie gegen inhärente Schwachstellen und Fehlverhalten und können diese anwenden, • können sie beurteilen, welche Kombination von Einzelkomponentenfehlern innerhalb welcher Zeitdauer zu Systemausfällen führen, • können sie Fragestellungen der Fehlervermeidung und Fehlerisolierung diskutieren sowie deren Vor- und Nachteile gegenüberstellen. Neben dem Erwerb dieser spezifischen Analyse-, Design- und Realisierungskompetenzen haben sie ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen und ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams vertieft.					
Inhalte • Ursachen für Funktionsbeeinträchtigungen und Systemversagen (Fehler, Ausfälle, Funktionsstörungen) • Qualitative und quantitative Kenngrößen zur Beurteilung von Fehlverhalten und Ausfallsicherheit • Aspekte und Einflüsse von Reparatur und kontinuierlicher Wartung (Ausfall- und Reparaturzeiten eines reparierbaren Systems) • Zuverlässigkeits-Zustandsübergangsmodelle und deren mathematische Behandlung (Zustandswahrscheinlichkeiten im Markov-Modell) • Restlebensdauer nach Teilausfällen und einfache Erneuerungsprozesse (Erneuerungsfunktion, Rekurrenzzeit, Funktionsprüfungen, Reservebetrieb) • Entwicklungswerkzeuge und Simulatoren (Prüf- und Diagnosetechniken, Standards)					
Literatur Börcsök, J.: Elektronische Sicherheitssysteme, Hüthig, 2007 Schneeweiss, W. G.: Zuverlässigkeitstechnik, Datakontext-Verlag, 1992 Biolini, A: Qualität und Zuverlässigkeit technischer Systeme, Springer-Verlag, 1999 Störmer, H.: Mathematische Theorie der Zuverlässigkeit, Oldenbourg Verlag, 1983					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter als PDF					

Modulbezeichnung / Prüfungsfach IT-Management für Software-Entwickler		Liste VM2	Modulnummer 8650	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 150 h Praktische Tätigkeit (Projekt) • 60 h Vor- und Nachbereitung (Theorieteil) • 90 h Anwesenheit (Vorlesung und Praktikum)			Dozenten Kröger, Schmid		Verantwortlich Kröger
Inhaltliche Voraussetzungen • Grundlagen der Verteilten Systeme • Grundlagen Rechnernetze • Grundlagen Softwaretechnik					
Lernziele Die Entwicklung komplexer verteilter IT-Systeme und unternehmenskritischer Anwendungen mit dem Ziel des Betriebs unter geforderten Qualitätsgütemerkmalen ist eine schwierige Aufgabe mit besonders hoher Bedeutung für die Praxis. Nach Abschluss des Moduls sollen die Studierenden erzielt haben: • Vertiefung des Fachwissens und der technologischen Kompetenzen bis zum Stand der Forschung auf dem Gebiet des IT-Managements • Vertiefung der Analyse- und Design-Kompetenz für Problemstellungen des Managements verteilter Systeme und Anwendungen einschl. Nutzung geeigneter mathematischer und Informatik-Modelle • Entwicklung eines ausgeprägten Urteilsvermögens für methodische und praktische Aspekte des Managements verteilter Systeme und Anwendungen • Vertiefung fachübergreifender Kompetenzen durch Betrachtung der Wechselwirkungen mit Geschäftsprozessen und betriebswirtschaftlichen Aspekten wie z.B. Accounting • Entwicklung von Fertigkeiten im Einsatz von Monitoring- und Management-Technologien bei der Entwicklung von Unternehmensanwendungen Neben diesen spezifischen Analyse-, Design- und Realisierungskompetenzen sowie spezifischen technologischen Kompetenzen haben die Studierenden vertieft: • ihre kommunikative Kompetenz durch Präsentation von eigenen Projektergebnissen • ihre sozialen Kompetenzen durch Arbeit in kleinen Projektteams					
Inhalte • Architekturen vernetzter Systeme und Anwendungen • Grundlagen des IT-Managements • Managementarchitekturen und ihre Sichten, verbreitete Managementarchitekturen • Standardisierte Vorgehensweisen für das Enterprise Management (z.B. ITIL, ISO) • Managementwerkzeuge und -plattformen • IT Service Management • Überblick über Methoden zur Leistungsbewertung • Instrumentierung von Systemen und Anwendungen von Entwurf bis Laufzeit • Automatisierung von IT-Management-Prozessen • Service Level Management • Selbstmanagement (Self-X) • Entwicklungswerkzeuge • Ausgewählte Beispiele und Lösungen, sowie aktuelle Forschungsthemen • Strukturierte selbstorganisierte Durchführung eines Projektes (Konzeption, Detailentwurf, technische Realisierung, Test, Bewertung, Projektpräsentation)					
Literatur Hegering, Abeck, Neumair: Integriertes Management vernetzter Systeme, dpunkt-Verlag, 1999 Beims: IT-Service Management in der Praxis mit ITIL3: Zielfindung, Methoden, Realisierung, Hanser, 2009 Keller: IT-Unternehmensarchitektur: Von der Geschäftsstrategie zur optimalen IT-Unterstützung, dpunkt, 2007 Jain: The art of computer systems performance analysis, Wiley, 1991 Projektspezifische Spezialliteratur					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Projektaufgaben (als pdf-Dateien) • ggfs Exkursion					

Modulbezeichnung / Prüfungsfach Prozessdatenverarbeitung		Liste VM2	Modulnummer 8660	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Kaiser, Linn		Verantwortlich Kaiser
Inhaltliche Voraussetzungen • Programmierkenntnisse in C • Grundlagen Rechnernetze, Betriebssysteme, verteilte Systeme/Microcomputer					
Lernziele Nach erfolgreichem Abschluss dieses Moduls sind die Studierenden in die Lage versetzt, • die grundlegenden Prinzipien und Wirkungsweisen eines Reglers verstehen und anwenden zu können • mit der Funktionsweise und den besonderen Eigenschaften von Feldbussen und Sensornetzen in Automatisierungsanwendungen vertraut zu sein • (Muster-)Beispiele von Implementierungen z.B. Klimatechnik, Hochregallager, Gebäudesystemtechnik gesehen zu haben • Methoden der Validierung von Automatisierungssoftware und daraus resultierende Konsequenzen zu kennen • Fachübergreifendes Verständnis z.B. für Fertigungs- und Logistik-Prozesse entwickelt zu haben • Die Auswirkung von Automatisierungslösungen unter ökonomischen und gesellschaftlichen Aspekten erkennen zu können Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von spezifischen Analyse-, Design- und Realisierungskompetenzen, zur Erweiterung von spezifischen technologischen Kompetenzen und zu fachübergreifenden technischen Systemkenntnissen bei.					
Inhalte • Grundlagen der Regelungstechnik und Systemtheorie • Praktische Programmierung von einfachen Microcontrollern z.B. Atmel • Interrupt-Handling • Struktur einer typischen digitalen Regelung • Messwerterfassung und -Ausgabe und das Timing • Signalaufbereitung, -filterung • Regelalgorithmen, Prinzipien der digitalen Regelung • Spezielle Automatisierungs-Komponenten (SPS, PLS, Netzadapter, FUP, KOP) • SCADA-Systeme • Grundlagen der Prozessvisualisierung • ggfs. Validierung von Steuerungs-Systemen • Eigene Projekte aus dem Bereich der Prozessdatenverarbeitung, Realisierung mit einfachen Microcontrollern oder Laptops, z.B. inverses Pendel, einfache Roboter, Steuerung / Visualisierung einer Solaranlage.					
Literatur U.Rembold/P.Levi: Realzeitsysteme zur Prozessautomatisierung, Hanser Verlag, 1994 Jan Lunze: Automatisierungstechnik. 2. Auflage. Oldenbourg Wissenschaftsverlag, 2007 K.W. Früh, U. Maier: Handbuch der Prozessautomatisierung. 4. Auflage. Oldenbourg Industrieverlag, 2009					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien) • Projektdurchführungen mit Microcontrollern • ggfs Exkursionen zu interessanten Projekten (Kläranlage, HRL, Pharmafertigung)					

Modulbezeichnung / Prüfungsfach Advanced Operating Systems		Liste VM2	Modulnummer 8670	Sem. 1, 2, 3	Studiengang Informatik (M.Sc.)
Credits / SWS 10 cp, 300 h 6 SWS	Lehrformen • Seminar (2 SWS) • Praktikum (4 SWS)		Häufigkeit bei Nachfrage	Bewertung • Klausur oder mündliche Prüfung oder praktische Tätigkeit und Fachgespräch (100% als PL)	
Aufwand • 90 h Anwesenheit (Vorlesung und Praktikum) • 60 h Vor- und Nachbereitung (Theorieteil) • 150 h Praktische Tätigkeit (Projekt)			Dozenten Kaiser, Kröger		Verantwortlich Kaiser
Inhaltliche Voraussetzungen • Programmieren in C • Betriebssysteme: Schnittstellen und prinzipieller Aufbau, Maschinensprache, Rechnerorganisation					
Lernziele Die Studierenden entwickeln ein eingehendes Verständnis über Implementierungstechniken und aktuelle Forschungsthemen im Bereich moderner Betriebssystemtechnologien. Dazu werden sie im Rahmen interaktiver Lehrveranstaltungen über spezielle Teilgebiete und durch die Analyse ausgewählter, aktueller Beiträge internationaler Tagungen mit den fortgeschrittenen Themen der Betriebssystementwicklung konfrontiert. Zudem besteht ein beträchtlicher Teil der Lehrveranstaltung in einem Projekt zur praktischen Konstruktion eines Betriebssystems, bei dem die Studierenden ihre erworbenen Kenntnisse und Fähigkeiten zum Einsatz bringen. Beide Komponenten zusammen geben den Studierenden eine solide theoretische Grundlage im Bereich moderner Betriebssysteme, die durch die praktische Anwendung weiter vertieft wird. Advanced operating systems befähigt Studierende, sich auf das Arbeitsgebiet der Betriebssystementwicklung zu spezialisieren und sowohl als Entwickler von Betriebssystemen oder eingebetteten Systemen als auch als Forscher in diesen Bereichen tätig zu werden. Die Lehrveranstaltung trägt zu den durch den Abschluss erreichten Kompetenzen in folgender Weise bei: • Durch das anspruchsvolle Projekt werden Studierende angeleitet, ihre analytischen Fähigkeiten im Umgang mit komplexen Systemen weiter zu entwickeln, kreativ Problemlösungen zu erarbeiten und gemeinschaftlich eine Anwendung ihrer erlernten Kenntnisse im Bereich der Betriebssysteme zu entwickeln. • Regelmäßig stattfindende Demonstrationen tragen zur Verbesserung der kommunikativen Kompetenzen bei. • Interaktive Untersuchung relevanter Forschungsansätze während der Lehrveranstaltung fördert die Entwicklung kritischer Analysefähigkeit. • Durch das Projekt werden Studierende auch in Ihrer Fähigkeit gefordert, Informationen zu recherchieren, deren Relevanz für das Projekt einzuschätzen, sie ggf. zu verarbeiten und ihre eigene Fähigkeit zum Verständnis der Informationen zu beurteilen.					
Inhalte Umfassender Einblick in Konzepte und Problemstellungen moderner Betriebssysteme, wie z.B.: • Mikrokerne und Interprozesskommunikation, • Betriebssystem-Server auf Anwendungsebene, • Entwurf und Implementierung mikrokernbasierter Systeme, • Performance, • Entwurf und Implementierung von Betriebssystemkernen, • Gerätetreiber, • Virtualisierung und Hypervisor, • Prozessplanung und Echtzeitfähigkeit, • Symmetrische Mehrprozessorsysteme und Hardware-Multithreading, • Einflüsse durch und Beeinflussung von Hardware-Caches, • Speicherschutz und Sicherheit, • Betriebssystementwürfe und daraus resultierende Probleme, • Aktuelle Themen der Betriebssystemforschung.					
Literatur A. Tanenbaum, A. Woodhull: Operating Systems: Design and Implementation, 2nd ed. 1997, Prentice Hall. W. Stallings: Operating Systems: Internals and Design Principles, 5th ed., 2004, Prentice Hall					
Medienform / Unterrichtssprache • Veranstaltungsspezifische Web-Seite • Skript/Folien und Übungsblätter (als pdf-Dateien)					