

---

# **Modulhandbuch**

**M. Sc. Informatik, PO 2017**

**Fakultät für Angewandte Informatik**

**Sommersemester 2017**

**Studienbeginn ab SS 2017**

---

Liebe Studierenden,

damit man bei unseren Studiengängen den Überblick behält, wer in welcher Prüfungsordnungs-version studiert und für wen somit welches Modulhandbuch gilt, gibt es nun für alle Informatik-Studiengänge (außer B.Sc. WIN, M.Sc. GeoInf, Elite-SE) eine Zusatzinformation auf der ersten Seite: Die eindeutige Angabe „Studienbeginn bis/ab“, z.B. „Studienbeginn bis SS16“ gilt für jene Studierenden, die sich spätestens zum SS 2016 im jeweiligen Studiengang eingeschrieben haben. Alle, die zum WS 2016/2017 begonnen haben, müssen das dazu gehörige Modulhandbuch aufschlagen.

So ist auf einen Blick erkennbar, ob ihr das richtige Modulhandbuch gefunden habt. Wenn es noch keine Angabe „bis“ gibt, so ist diese noch offen, da es (noch) keine neuere Ausprägung des Studiengangs gibt (derzeit gibt es nur eine Version z.B. im B.Sc. GeoInf und B.Sc./M.Sc. IngInf).

Im M.Sc. Informatik sowie im M.Sc. Informatik und Multimedia gibt es eine neue Prüfungsordnungs-version und somit nun jeweils zwei Modulhandbücher: Eines für alle Studierenden, die ihr Studium bis zum WS 2016/2017 begonnen haben und eines für alle, die ab dem SS 2017 beginnen.

Die wichtigsten Änderungen sind folgende:

- Statt eines Notendurchschnitts von 3,0 im Bachelor muss man mindestens 2,7 erreicht haben
- Dabei werden noch nicht abgelegte/geprüfte Module als 4,0 bewertet angerechnet
- Schlüsselqualifikationen sind nicht mehr verpflichtend im Umfang von 6 LP zu erbringen, sondern freiwillig bis max. 2 LP

Da das Modulhandbuch ein Service für euch als Studierende ist, arbeite ich eng mit der Studierendenvertretung Informatik zusammen. Solltet Ihr Anregungen, Fragen, Kritik oder Verbesserungsvorschläge zum Modulhandbuch haben, so teilt diese einfach der Studierendenvertretung Informatik mit. Ihr erreicht sie unter

[fsinfo@informatik.uni-augsburg.de](mailto:fsinfo@informatik.uni-augsburg.de)

und persönlich im Raum 1007N.

Viele Grüße,

Euer Modulhandbuch-Beauftragter

Martin Frieb

ID	Modul	Semester	ECTS	SWS	Prüfung
<b>M.Sc. Informatik (PO '17), Teilbereich Informatik</b>					
mindestens 90 Leistungspunkte aus dem Teilbereich Informatik; davon in zwei Schwerpunkten nach Abs. 1, außer dem Schwerpunkt Schlüsselqualifikation, mindestens 16 Leistungspunkte, in jedem weiteren der in Abs. 1 genannten Schwerpunkte, außer dem Schwerpunkt Schlüsselqualifikation, vier Leistungspunkte und höchstens zwei Leistungspunkte aus dem Schwerpunkt Schlüsselqualifikation; innerhalb der 90 Leistungspunkte nach Halbsatz 1 sind genau ein Projektmodul mit zehn Leistungspunkten sowie mindestens ein Seminarmodul mit vier Leistungspunkten, aber maximal zwei Seminarmodule, zu erbringen; (§5 (2) der PO)					
<b>1</b>	<b>Modulgruppe: Softwaretechnik und Programmiersprachen</b>				
INF-0031	Compilerbau	jedes Sommersemester	6	3 Vorlesung 2 Übung	Klausur 90Minuten
INF-0032	Automotive Software Engineering	unregelmäßig	5	3 Vorlesung	Mündliche Prüfung 30Minuten
INF-0033	Modellgetriebene Softwareentwicklung	jedes Sommersemester	6	3 Vorlesung 2 Übung	Klausur 90Minuten
INF-0034	Softwarearchitekturen und Enterprise Architecture Management	jedes Wintersemester	6	3 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0035	Agile Methoden	jedes Wintersemester	6	3 Vorlesung 2 Übung	Klausur 60Minuten
INF-0036	Praktikum Business & Information Systems Engineering IV (MA)	unregelmäßig	6	6 Praktikum	Mündliche Prüfung 30Minuten
INF-0037	Praktikum Automotive Software Engineering	jedes Semester	6	6 Praktikum	Mündliche Prüfung 30Minuten
INF-0038	Praktikum Avionic Software Engineering	unregelmäßig	6	6 Praktikum	Mündliche Prüfung 30Minuten

---

INF-0039	Seminar über Software Engineering verteilter Systeme (MA)	jedes Semester	4	2 Seminar	Seminar
INF-0040	Seminar Grundlagen des Software Engineering für Automotive Systems (MA)	jedes Semester	4	2 Seminar	Seminar
INF-0041	Seminar Grundlagen des Software Engineering für Avionic Systems (MA)	jedes Semester	4	2 Seminar	Seminar
INF-0042	Projektmodul Softwaremethodiken für verteilte Systeme	nach Bedarf	10	1 Praktikum	Praktikum
INF-0066	Organic Computing II	jedes Sommersemester	5	2 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0068	Interactive Simulation	letzte Prüfung: SS 2017	5	2 Vorlesung 2 Übung	Hausarbeit/Seminararbeit Mündliche Prüfung 15Minuten
INF-0070	Seminar Organic Computing	jedes Wintersemester	4	2 Seminar	Seminar
INF-0071	Seminar Naturanaloge Algorithmen und Multiagentensysteme	jedes Sommersemester	4	2 Seminar	Seminar
INF-0072	Projektmodul Organic Computing	nach Bedarf	10	1 Praktikum	Praktikum
INF-0108	Projektmodul Lehrprofessur für Informatik	nach Bedarf	10	1 Praktikum	Praktikum
INF-0129	Softwaretechnik II	jedes Wintersemester	8	2 Vorlesung 4 Übung	Klausur 90Minuten

---

---

INF-0130	Formale Methoden im Software Engineering	jedes Sommersemester	8	2 Vorlesung 4 Übung	Mündliche Prüfung 45Minuten
INF-0131	Software- und Systemsicherheit	jedes Wintersemester	8	2 Vorlesung 4 Übung	Mündliche Prüfung 45Minuten
INF-0133	Selbstorganisierende, adaptive Systeme	jedes Wintersemester	8	2 Vorlesung 4 Übung	Mündliche Prüfung 45Minuten
INF-0136	Seminar Software- und Systems Engineering (Master)	jedes Semester	4	2 Seminar	Seminar
INF-0137	Projektmodul Software- und Systems Engineering	nach Bedarf	10	1 Praktikum	Praktikum
INF-0177	Einführung in die Künstliche Intelligenz	wird nicht mehr angeboten!	5	2 Vorlesung 2 Übung	Klausur
INF-0189	Qualitätssicherung im Software Engineering	unregelmäßig (i. d. R. im SoSe)	5	2 Vorlesung 2 Übung	Klausur 90Minuten
INF-0217	Praktikum Autonomes Fahren	jedes Semester	10	10 Praktikum	Mündliche Prüfung 30Minuten
INF-0219	Seminar Architektur- und Technologiekonzepte (MA)	unregelmäßig	4	2 Seminar	Seminar
INF-0232	Seminar Medical Information Sciences (MA)	jedes Semester	4	2 Seminar	Seminar
INF-0233	Industrierobotik	jedes Semester	8	2 Vorlesung 4 Übung	praktische Prüfung 45Minuten

---

INF-0235	Software für Industrie 4.0	jedes Wintersemester	5	2 Vorlesung 2 Übung	Mündliche Prüfung 45Minuten
INF-0248	Kollaborative Robotik	unregelmäßig	8	2 Vorlesung 4 Übung	praktische Prüfung 60Minuten
<b>2</b>	<b>Modulgruppe: Datenbanken und Informationssysteme</b>				
INF-0077	Suchmaschinen	unregelmäßig (i. d. R. im SoSe)	8	4 Vorlesung 2 Übung	Klausur 90Minuten
INF-0078	Datenbankprogrammierung (Oracle)	unregelmäßig	5	2 Vorlesung 2 Übung	Klausur 60Minuten
INF-0080	Projektmodul Datenbanken und Informationssysteme	nach Bedarf	10	1 Praktikum	Praktikum
INF-0092	Multimedia II: Machine Learning and Computer Vision	jedes Sommersemester	8	4 Vorlesung 2 Übung	Klausur 120Minuten
INF-0117	Funktionale Modellierung für Geoinformationssysteme	unregelmäßig	5	2 Vorlesung 2 Übung	Klausur 120Minuten
INF-0118	Seminar Programmiermethodik und Multimediale Informationssysteme für Master	in der Regel mind. 1x pro Studienjahr	4	2 Seminar	Seminar
INF-0119	Projektmodul Programmiermethodik und Multimediale Informationssysteme	nach Bedarf	10	1 Praktikum	Praktikum
INF-0213	Parallele und Verteilte Datenbanksysteme	unregelmäßig	5	2 2	Mündliche Prüfung 30Minuten

INF-0227	Seminar Datenbanksysteme für Master	unregelmäßig (i. d. R. im SoSe)	4	2 Seminar	Seminar Stunden
INF-0240	Seminar Informationssysteme für Master	unregelmäßig (i. d. R. im WS)	4	2 Seminar	Seminar Stunden
<b>3</b>	<b>Modulgruppe: Rechnerkommunikation und Systemnahe Informatik</b>				
INF-0067	Peer-to-Peer und Cloud Computing	jedes Wintersemester	5	2 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0069	Weiterführende Betriebssystemkonzepte	unregelmäßig	8	2 Vorlesung 4 Übung	Klausur 90Minuten
INF-0071	Seminar Naturanaloge Algorithmen und Multiagentensysteme	jedes Sommersemester	4	2 Seminar	Seminar
INF-0084	Seminar Next Generation Networks	jedes Sommersemester	4	2 Seminar	Seminar
INF-0085	Projektmodul Kommunikationssysteme	nach Bedarf	10	1 Praktikum	Praktikum
INF-0145	Mikrorechnertechnik und Echtzeitsysteme	jedes Wintersemester	6	3 Vorlesung 1 Übung	Klausur 60Minuten
INF-0146	Cyber-Physical Systems	wird nicht mehr angeboten!	6	3 Vorlesung 1 Übung	Klausur 90Minuten
INF-0147	Prozessorarchitektur	jedes Sommersemester	5	2 Vorlesung 2 Übung	Klausur 60Minuten

INF-0148	Entwurf und Analyse fehlertolerierender Rechensysteme	jedes Wintersemester	5	2 Vorlesung 2 Übung	Klausur 60Minuten
INF-0149	Praktikum Eingebettete Systeme	jedes Sommersemester	5	4 Praktikum	Praktikum
INF-0150	Hardware-Entwurf	jedes Wintersemester	8	2 Vorlesung 4 Übung	Praktikum
INF-0151	Praktikum Multicore-Programmierung	wurde ersetzt durch INF-0216	5	4 Praktikum	Praktikum
INF-0152	Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen	jedes Sommersemester	4	2 Seminar	Seminar
INF-0153	Seminar Safety-Critical Systems	jedes Wintersemester	4	2 Seminar	Seminar
INF-0154	Projektmodul Systemnahe Informatik und Kommunikationssysteme	nach Bedarf	10	1 Praktikum	Praktikum
INF-0216	Vertiefte Multicore-Programmierung	jedes Sommersemester	8	2 Vorlesung 4 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten) Stunden
<b>4</b>	<b><i>Modulgruppe: Theoretische Informatik</i></b>				
INF-0050	Constrained data structures	unregelmäßig	4	3 Vorlesung 1 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0051	Algorithmen für NP-harte Probleme	unregelmäßig	8	4 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)



---

INF-0052	Einführung in die Komplexitätstheorie	unregelmäßig	5	2 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0053	I/O-effiziente Algorithmen	unregelmäßig	5	2 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0054	Datenstrukturen	unregelmäßig	8	4 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0055	Teile-und-Herrsche-Algorithmen	unregelmäßig	5	2 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0056	Online-Algorithmen	unregelmäßig	5	2 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0057	Praktikum: NP-harte Graphprobleme	unregelmäßig	8	6 Praktikum	Praktikum
INF-0058	Seminar Algorithmen und Datenstrukturen für Master	unregelmäßig	4	2 Seminar	Seminar
INF-0059	Projektmodul Theoretische Informatik	nach Bedarf	10	1 Praktikum	Praktikum
INF-0107	Seminar Petrinetze	unregelmäßig	4	2 Seminar	Seminar
INF-0108	Projektmodul Lehrprofessur für Informatik	nach Bedarf	10	1 Praktikum	Praktikum
INF-0116	Algebraische Semantik und Algebraische Systementwicklung	unregelmäßig	8	4 Vorlesung 2 Übung	Klausur 120Minuten

---

---

INF-0118	Seminar Programmiermethodik und Multimediale Informationssysteme für Master	in der Regel mind. 1x pro Studienjahr nach Bedarf	4	2 Seminar	Seminar
INF-0119	Projektmodul Programmiermethodik und Multimediale Informationssysteme		10	1 Praktikum	Praktikum
INF-0130	Formale Methoden im Software Engineering	jedes Sommersemester	8	2 Vorlesung 4 Übung	Mündliche Prüfung 45Minuten
INF-0156	Algebraische Beschreibung paralleler Prozesse	unregelmäßig	6	3 Vorlesung 1 Übung	Mündliche Prüfung 30Minuten
INF-0157	Endliche Automaten	unregelmäßig	5	3 Vorlesung + Übung	Mündliche Prüfung 30Minuten
INF-0161	Petrinetze - eine Theorie paralleler Systeme	unregelmäßig (i. d. R. im WS)	5	2 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0163	Verteilte Algorithmen	unregelmäßig	8	4 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0164	Seminar Theorie verteilter Systeme A	unregelmäßig	4	2 Seminar	Seminar
INF-0165	Projektmodul Theorie verteilter Systeme	nach Bedarf	10	1	Projektarbeit
INF-0201	Platzeffiziente Algorithmen	unregelmäßig	5	2 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0243	Process Mining	jedes Sommersemester	6	3 Vorlesung 1 Übung	Klausur 90Minuten

---

<b>5 Modulgruppe: Multimedia</b>						
INF-0068	Interactive Simulation	letzte Prüfung: SS 2017	5	2 Vorlesung 2 Übung	Hausarbeit/Seminararbeit Mündliche Prüfung 15Minuten	
INF-0088	Bayesian Networks	jedes Sommersemester	5	2 Vorlesung 2 Übung	Klausur 90Minuten	
INF-0092	Multimedia II: Machine Learning and Computer Vision	jedes Sommersemester	8	4 Vorlesung 2 Übung	Klausur 120Minuten	
INF-0093	Probabilistic Robotics	jedes Wintersemester	5	2 Vorlesung 2 Übung	Klausur 90Minuten	
INF-0094	Maschinelles Lernen	unregelmäßig	5	2 Vorlesung 2 Übung	Klausur 90Minuten	
INF-0095	Seminar Multimedia Computing (MA)	jedes Sommersemester	4	2 Seminar	Seminar	
INF-0096	Projektmodul Multimedia Computing	nach Bedarf	10	1 Praktikum	Praktikum	
INF-0112	Graphikprogrammierung	unregelmäßig	8	4 Vorlesung 2 Übung	Klausur 120Minuten	
INF-0118	Seminar Programmiermethodik und Multimediale Informationssysteme für Master	in der Regel mind. 1x pro Studienjahr	4	2 Seminar	Seminar	
INF-0119	Projektmodul Programmiermethodik und Multimediale Informationssysteme	nach Bedarf	10	1 Praktikum	Praktikum	

---

INF-0170	Projektmodul Human-Centered Multimedia	nach Bedarf	10	1 Praktikum	Praktikum
INF-0175	Multimedia I: Usability Engineering	jedes Wintersemester	8	4 Vorlesung 2 Übung	Übung + Praktikum
INF-0176	Digital Signal Processing II	wird nicht mehr angeboten!	6	4 Vorlesung	Klausur 100Minuten
INF-0177	Einführung in die Künstliche Intelligenz	wird nicht mehr angeboten!	5	2 Vorlesung 2 Übung	Klausur
INF-0178	Praktikum Usability Engineering	jedes Sommersemester	8	6 Praktikum	Projektarbeit
INF-0179	Einführung in die Spieleprogrammierung	jedes Sommersemester	8	2 Vorlesung 4 Übung	Projektarbeit
INF-0180	Computational Intelligence	jedes Wintersemester	8	2 Vorlesung 4 Übung	Mündliche Prüfung
INF-0181	Praktikum Multimodal Interaction	jedes Semester	8	6 Praktikum	Projektarbeit
INF-0182	Praktikum Multimodale Echtzeitsignalverarbeitung	jedes Sommersemester	8	6 Praktikum	Projektarbeit
INF-0183	Praktikum Spieleprogrammierung	jedes Wintersemester	8	6 Praktikum	Projektarbeit
INF-0184	Seminar User Interface Design	jedes Wintersemester	4	2 Seminar	Seminar

---

INF-0185	Seminar Advanced Topics in Signal and Pattern Recognition	wird nicht mehr angeboten!	4	2 Seminar	Seminar
INF-0198	Intelligente Systeme	unregelmäßig	5	2 Vorlesung 2 Übung	Klausur 90Minuten
INF-0207	Reinforcement Learning	jedes Wintersemester	8	2 Vorlesung 4 Übung	Projektarbeit (Projektarbeit / mündliche Prüfung) 30Minuten
INF-0250	Praktikum Reinforcement Learning	jedes Sommersemester	8	6 Praktikum	Projektarbeit Stunden
INF-0251	Seminar Artificial Intelligence	in der Regel mind. 1x pro Studienjahr	4	2 Seminar	Seminar Stunden
<b>6</b>	<b>Modulgruppe: Schlüsselqualifikation</b>		<b>0 - 2</b>		
ZCS-2000	Softskills	jedes Semester	2	2 Kurs	Beteiligungsnachweis Minuten

---

ID	Modul	Semester	ECTS	SWS	Prüfung
	<b>Mastermodul</b> 30 Leistungspunkte im Rahmen des Mastermoduls				
INF-0003	Masterarbeit	nach Bedarf	30	1	Masterarbeit

---

ID	Modul	Semester	ECTS	SWS	Prüfung
	<b>Freiwillige Veranstaltungen</b>				
	Die hier aufgeführten Veranstaltungen sind freiwillig und geben keine Leistungspunkte. Ihre Inhalte sind jedoch eine sinnvolle Ergänzung zum bestehenden Lehrangebot.				
INF-0221	Anleitung zu wissenschaftlichen Arbeiten	jedes Semester	0	1	
INF-0222	Oberseminar Informatik	jedes Semester	0	2 Seminar	

# Übersicht nach Modulgruppen

## 1) M.Sc. Informatik (PO '17), Teilbereich Informatik ECTS: 88 - 90

mindestens 90 Leistungspunkte aus dem Teilbereich Informatik; davon in zwei Schwerpunkten nach Abs. 1, außer dem Schwerpunkt Schlüsselqualifikation, mindestens 16 Leistungspunkte, in jedem weiteren der in Abs. 1 genannten Schwerpunkte, außer dem Schwerpunkt Schlüsselqualifikation, vier Leistungspunkte und höchstens zwei Leistungspunkte aus dem Schwerpunkt Schlüsselqualifikation; innerhalb der 90 Leistungspunkte nach Halbsatz 1 sind genau ein Projektmodul mit zehn Leistungspunkten sowie mindestens ein Seminarmodul mit vier Leistungspunkten, aber maximal zwei Seminarmodule, zu erbringen; (§5 (2) der PO)

### a) Softwaretechnik und Programmiersprachen

INF-0031: Compilerbau (6 ECTS/LP, Wahlpflicht).....	7
INF-0032: Automotive Software Engineering (5 ECTS/LP, Wahlpflicht).....	8
INF-0033: Modellgetriebene Softwareentwicklung (6 ECTS/LP, Wahlpflicht).....	9
INF-0034: Softwarearchitekturen und Enterprise Architecture Management (6 ECTS/LP, Wahlpflicht).....	10
INF-0035: Agile Methoden (6 ECTS/LP, Wahlpflicht).....	12
INF-0036: Praktikum Business & Information Systems Engineering IV (MA) (6 ECTS/LP, Wahlpflicht).....	13
INF-0037: Praktikum Automotive Software Engineering (6 ECTS/LP, Wahlpflicht).....	14
INF-0038: Praktikum Avionic Software Engineering (6 ECTS/LP, Wahlpflicht).....	16
INF-0039: Seminar über Software Engineering verteilter Systeme (MA) (4 ECTS/LP, Wahlpflicht).....	18
INF-0040: Seminar Grundlagen des Software Engineering für Automotive Systems (MA) (4 ECTS/LP, Wahlpflicht).....	19
INF-0041: Seminar Grundlagen des Software Engineering für Avionic Systems (MA) (4 ECTS/LP, Wahlpflicht).....	20
INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme (10 ECTS/LP, Wahlpflicht).....	21
INF-0066: Organic Computing II (5 ECTS/LP, Wahlpflicht).....	22
INF-0068: Interactive Simulation (5 ECTS/LP, Wahlpflicht).....	24
INF-0070: Seminar Organic Computing (4 ECTS/LP, Wahlpflicht).....	26
INF-0071: Seminar Naturalanaloge Algorithmen und Multiagentensysteme (4 ECTS/LP, Wahlpflicht).....	27
INF-0072: Projektmodul Organic Computing (10 ECTS/LP, Wahlpflicht).....	28
INF-0108: Projektmodul Lehrprofessur für Informatik (10 ECTS/LP, Wahlpflicht).....	29



INF-0129: Softwaretechnik II (8 ECTS/LP, Wahlpflicht).....	31
INF-0130: Formale Methoden im Software Engineering (8 ECTS/LP, Wahlpflicht).....	33
INF-0131: Software- und Systemsicherheit (8 ECTS/LP, Wahlpflicht).....	35
INF-0133: Selbstorganisierende, adaptive Systeme (8 ECTS/LP, Wahlpflicht).....	37
INF-0136: Seminar Software- und Systems Engineering (Master) (4 ECTS/LP, Wahlpflicht).....	39
INF-0137: Projektmodul Software- und Systems Engineering (10 ECTS/LP, Wahlpflicht).....	41
INF-0177: Einführung in die Künstliche Intelligenz (5 ECTS/LP, Wahlpflicht).....	42
INF-0189: Qualitätssicherung im Software Engineering (5 ECTS/LP, Wahlpflicht).....	43
INF-0217: Praktikum Autonomes Fahren (10 ECTS/LP, Wahlpflicht).....	45
INF-0219: Seminar Architektur- und Technologiekonzepte (MA) (4 ECTS/LP, Wahlpflicht).....	47
INF-0232: Seminar Medical Information Sciences (MA) (4 ECTS/LP, Wahlpflicht).....	48
INF-0233: Industrierobotik (8 ECTS/LP, Wahlpflicht).....	49
INF-0235: Software für Industrie 4.0 (5 ECTS/LP, Wahlpflicht).....	51
INF-0248: Kollaborative Robotik (8 ECTS/LP, Wahlpflicht).....	53

## **b) Datenbanken und Informationssysteme**

INF-0077: Suchmaschinen (8 ECTS/LP, Wahlpflicht).....	55
INF-0078: Datenbankprogrammierung (Oracle) (5 ECTS/LP, Wahlpflicht).....	57
INF-0080: Projektmodul Datenbanken und Informationssysteme (10 ECTS/LP, Wahlpflicht).....	58
INF-0092: Multimedia II: Machine Learning and Computer Vision (8 ECTS/LP, Wahlpflicht).....	59
INF-0117: Funktionale Modellierung für Geoinformationssysteme (5 ECTS/LP, Wahlpflicht).....	61
INF-0118: Seminar Programmiermethodik und Multimediale Informationssysteme für Master (4 ECTS/LP, Wahlpflicht).....	63
INF-0119: Projektmodul Programmiermethodik und Multimediale Informationssysteme (10 ECTS/LP, Wahlpflicht).....	64
INF-0213: Parallele und Verteilte Datenbanksysteme (5 ECTS/LP, Wahlpflicht).....	65
INF-0227: Seminar Datenbanksysteme für Master (4 ECTS/LP, Wahlpflicht).....	67
INF-0240: Seminar Informationssysteme für Master (4 ECTS/LP, Wahlpflicht).....	69

## **c) Rechnerkommunikation und Systemnahe Informatik**

INF-0067: Peer-to-Peer und Cloud Computing (5 ECTS/LP, Wahlpflicht).....	70
INF-0069: Weiterführende Betriebssystemkonzepte (8 ECTS/LP, Wahlpflicht).....	72

INF-0071: Seminar Naturanaloge Algorithmen und Multiagentensysteme (4 ECTS/LP, Wahlpflicht).....	74
INF-0084: Seminar Next Generation Networks (4 ECTS/LP, Wahlpflicht).....	75
INF-0085: Projektmodul Kommunikationssysteme (10 ECTS/LP, Wahlpflicht).....	76
INF-0145: Mikrorechnerarchitektur und Echtzeitsysteme (6 ECTS/LP, Wahlpflicht).....	77
INF-0146: Cyber-Physical Systems (6 ECTS/LP, Wahlpflicht).....	79
INF-0147: Prozessorarchitektur (5 ECTS/LP, Wahlpflicht).....	81
INF-0148: Entwurf und Analyse fehlertolerierender Rechensysteme (5 ECTS/LP, Wahlpflicht).....	83
INF-0149: Praktikum Eingebettete Systeme (5 ECTS/LP, Wahlpflicht).....	85
INF-0150: Hardware-Entwurf (8 ECTS/LP, Wahlpflicht).....	86
INF-0151: Praktikum Multicore-Programmierung (5 ECTS/LP, Wahlpflicht).....	88
INF-0152: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen (4 ECTS/LP, Wahlpflicht).....	89
INF-0153: Seminar Safety-Critical Systems (4 ECTS/LP, Wahlpflicht).....	90
INF-0154: Projektmodul Systemnahe Informatik und Kommunikationssysteme (10 ECTS/LP, Wahlpflicht).....	91
INF-0216: Vertiefte Multicore-Programmierung (8 ECTS/LP, Wahlpflicht).....	92

## **d) Theoretische Informatik**

INF-0050: Constrained data structures (4 ECTS/LP, Wahlpflicht).....	94
INF-0051: Algorithmen für NP-harte Probleme (8 ECTS/LP, Wahlpflicht).....	96
INF-0052: Einführung in die Komplexitätstheorie (5 ECTS/LP, Wahlpflicht).....	98
INF-0053: I/O-effiziente Algorithmen (5 ECTS/LP, Wahlpflicht).....	99
INF-0054: Datenstrukturen (8 ECTS/LP, Wahlpflicht).....	101
INF-0055: Teile-und-Herrsche-Algorithmen (5 ECTS/LP, Wahlpflicht).....	102
INF-0056: Online-Algorithmen (5 ECTS/LP, Wahlpflicht).....	104
INF-0057: Praktikum: NP-harte Graphprobleme (8 ECTS/LP, Wahlpflicht).....	105
INF-0058: Seminar Algorithmen und Datenstrukturen für Master (4 ECTS/LP, Wahlpflicht).....	106
INF-0059: Projektmodul Theoretische Informatik (10 ECTS/LP, Wahlpflicht).....	107
INF-0107: Seminar Petrinetze (4 ECTS/LP, Wahlpflicht).....	108
INF-0108: Projektmodul Lehrprofessur für Informatik (10 ECTS/LP, Wahlpflicht).....	109
INF-0116: Algebraische Semantik und Algebraische Systementwicklung (8 ECTS/LP, Wahlpflicht).....	111

INF-0118: Seminar Programmiermethodik und Multimediale Informationssysteme für Master (4 ECTS/LP, Wahlpflicht).....	113
INF-0119: Projektmodul Programmiermethodik und Multimediale Informationssysteme (10 ECTS/LP, Wahlpflicht).....	114
INF-0130: Formale Methoden im Software Engineering (8 ECTS/LP, Wahlpflicht).....	115
INF-0156: Algebraische Beschreibung paralleler Prozesse (6 ECTS/LP, Wahlpflicht).....	117
INF-0157: Endliche Automaten (5 ECTS/LP, Wahlpflicht).....	119
INF-0161: Petrinetze - eine Theorie paralleler Systeme (5 ECTS/LP, Wahlpflicht).....	120
INF-0163: Verteilte Algorithmen (8 ECTS/LP, Wahlpflicht).....	121
INF-0164: Seminar Theorie verteilter Systeme A (4 ECTS/LP, Wahlpflicht).....	123
INF-0165: Projektmodul Theorie verteilter Systeme (10 ECTS/LP, Wahlpflicht).....	124
INF-0201: Platzeffiziente Algorithmen (5 ECTS/LP, Wahlpflicht).....	125
INF-0243: Process Mining (6 ECTS/LP, Wahlpflicht).....	127

## **e) Multimedia**

INF-0068: Interactive Simulation (5 ECTS/LP, Wahlpflicht).....	129
INF-0088: Bayesian Networks (5 ECTS/LP, Wahlpflicht).....	131
INF-0092: Multimedia II: Machine Learning and Computer Vision (8 ECTS/LP, Wahlpflicht).....	133
INF-0093: Probabilistic Robotics (5 ECTS/LP, Wahlpflicht).....	135
INF-0094: Maschinelles Lernen (5 ECTS/LP, Wahlpflicht).....	137
INF-0095: Seminar Multimedia Computing (MA) (4 ECTS/LP, Wahlpflicht).....	139
INF-0096: Projektmodul Multimedia Computing (10 ECTS/LP, Wahlpflicht).....	140
INF-0112: Graphikprogrammierung (8 ECTS/LP, Wahlpflicht).....	141
INF-0118: Seminar Programmiermethodik und Multimediale Informationssysteme für Master (4 ECTS/LP, Wahlpflicht).....	143
INF-0119: Projektmodul Programmiermethodik und Multimediale Informationssysteme (10 ECTS/LP, Wahlpflicht).....	144
INF-0170: Projektmodul Human-Centered Multimedia (10 ECTS/LP, Wahlpflicht).....	145
INF-0175: Multimedia I: Usability Engineering (8 ECTS/LP, Wahlpflicht).....	146
INF-0176: Digital Signal Processing II (6 ECTS/LP, Wahlpflicht).....	147
INF-0177: Einführung in die Künstliche Intelligenz (5 ECTS/LP, Wahlpflicht).....	148
INF-0178: Praktikum Usability Engineering (8 ECTS/LP, Wahlpflicht).....	149
INF-0179: Einführung in die Spieleprogrammierung (8 ECTS/LP, Wahlpflicht).....	151

INF-0180: Computational Intelligence (8 ECTS/LP, Wahlpflicht).....	153
INF-0181: Praktikum Multimodal Interaction (8 ECTS/LP, Wahlpflicht).....	155
INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung (8 ECTS/LP, Wahlpflicht).....	157
INF-0183: Praktikum Spieleprogrammierung (8 ECTS/LP, Wahlpflicht).....	159
INF-0184: Seminar User Interface Design (4 ECTS/LP, Wahlpflicht).....	160
INF-0185: Seminar Advanced Topics in Signal and Pattern Recognition (4 ECTS/LP, Wahlpflicht).....	161
INF-0198: Intelligente Systeme (5 ECTS/LP, Wahlpflicht).....	162
INF-0207: Reinforcement Learning (8 ECTS/LP, Wahlpflicht).....	163
INF-0250: Praktikum Reinforcement Learning (8 ECTS/LP, Wahlpflicht).....	164
INF-0251: Seminar Artificial Intelligence (4 ECTS/LP, Wahlpflicht).....	165

## **f) Schlüsselqualifikation ECTS: 0 - 2**

ZCS-2000: Softskills (2 ECTS/LP, Wahlfach).....	167
---	-----

## **2) Mastermodul ECTS: 30**

30 Leistungspunkte im Rahmen des Mastermoduls

INF-0003: Masterarbeit (30 ECTS/LP, Pflicht).....	170
---	-----

## **3) Freiwillige Veranstaltungen**

Die hier aufgeführten Veranstaltungen sind freiwillig und geben keine Leistungspunkte. Ihre Inhalte sind jedoch eine sinnvolle Ergänzung zum bestehenden Lehrangebot.

INF-0221: Anleitung zu wissenschaftlichen Arbeiten (0 ECTS/LP, Wahlfach).....	171
INF-0222: Oberseminar Informatik (0 ECTS/LP, Wahlfach).....	172

<b>Modul INF-0031: Compilerbau</b>		6 ECTS/LP
Version 2.0.0 (seit WS14/15) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Compilerbautechnologien verstehen, anwenden, bewerten, wissenschaftlich weiterentwickeln können.		
<b>Schlüsselqualifikationen:</b> Erlernen des eigenständigen Arbeitens mit Lehrbüchern (oder englischsprachiger Fachliteratur); Erwerb von Abstraktionsfähigkeiten.		
<b>Arbeitsaufwand:</b> Gesamt: 180 Std. 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 5	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

**Modulteile****Modulteil: Compilerbau (Vorlesung)****Lehrformen:** Vorlesung**Sprache:** Deutsch**SWS:** 3**Inhalte:**

In dieser Vorlesung werden wir uns mit der Übersetzung objektorientierter, funktionaler und logischer Programmiersprachen beschäftigen. Insbesondere werden dabei Smalltalk, C++ und Java, sowie Haskell und Prolog genauer untersucht.

**Literatur:**

- Aho et al: Compilerbau

**Modulteil: Compilerbau (Übung)****Lehrformen:** Übung**Sprache:** Deutsch**SWS:** 2**Prüfung****Compilerbau (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

<b>Modul INF-0032: Automotive Software Engineering</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an den Modulveranstaltungen ist der Studierende in der Lage die Software Engineering Methoden im Automotive Umfeld zu verstehen, anzuwenden und zu bewerten.		
<b>Schlüsselqualifikationen:</b> Erlernen des eigenständigen Arbeitens mit Lehrbüchern (oder englischsprachiger Fachliteratur), Erwerb von Abstraktionsfähigkeiten		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 52 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 53 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 45 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 3	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Automotive Software Engineering (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 3
<b>Inhalte:</b> Die Vorlesung beschäftigt sich mit allen Teilprozessen des Software-Engineerings und zeigt diese anhand von Beispielen aus dem Bereich Automotive: Projektmanagement, Risikomanagement, Qualitätssicherung, Konfigurationsmanagement, Änderungsmanagement, System Analyse, System Architektur, Software Design, Software Implementierung, Software Test sowie Zulieferer Management. Dabei wird auf Besonderheiten der Automotive Standards AUTOSAR und ISO26262 für sicherheitskritische Entwicklung eingegangen. In der Vorlesung werden Software-Entwicklungsprozesse von Automobilherstellern als auch von Automobilzulieferern exemplarisch gezeigt und diskutiert.
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Software Engineering nach Automotive SPICE: Entwicklungsprozesse in der Praxis: ein Continental-Projekt auf dem Weg zu Level 3 Holger Höhn, Bernhard Sechser, Klaudia Dussa-Zieger; 2009; Dpunkt Verlag</li> </ul>

<b>Prüfung</b> <b>Automotive Software Engineering (mündl. Prüfung)</b> Mündliche Prüfung / Prüfungsdauer: 30 Minuten
--

<b>Modul INF-0033: Modellgetriebene Softwareentwicklung</b>		6 ECTS/LP
Version 2.0.0 (seit WS14/15) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Die Teilnehmer der Vorlesung können die MDSO zugrunde liegenden Konzepte verstehen und anwenden. Sie besitzen einen Einblick in aktuelle Technologien und Standards für MDSO und können diese bewerten.		
<b>Schlüsselqualifikationen:</b> Erlernen des eigenständigen Arbeitens mit Lehrbüchern (oder englischsprachiger Fachliteratur); Erwerb von Abstraktionsfähigkeiten		
<b>Arbeitsaufwand:</b> Gesamt: 180 Std. 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 45 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b>		
keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 5	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Modellgetriebene Softwareentwicklung (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 3		
<b>Inhalte:</b>		
Modellgetriebene Softwareentwicklung oder Model Driven Software Development (MDSO) befasst sich mit der Effizienzsteigerung in der Softwareherstellung durch Automatisierung und Wiederverwendung. Dabei werden Infrastrukturcode, Subsysteme, Konfigurationen oder ganze Anwendungen aus Modellen generiert.		
<b>Literatur:</b>		
<ul style="list-style-type: none"> <li>• Folien</li> <li>• Pohl et al. Software Product Line Engineering: Foundations, Principles, and Techniques</li> <li>• Kleppe et al: MDA explained</li> <li>• Hitz et al: UML@Work</li> <li>• weitere Literatur in der Vorlesung zu speziellen Themen</li> </ul>		
<b>Modulteil: Modellgetriebene Softwareentwicklung (Übung)</b>		
<b>Lehrformen:</b> Übung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Prüfung</b>		
<b>Modellgetriebene Softwareentwicklung (Klausur)</b>		
Klausur / Prüfungsdauer: 90 Minuten		

<b>Modul INF-0034: Softwarearchitekturen und Enterprise Architecture Management</b>		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage SW-Architekturen zu erstellen, zu bewerten und zu dokumentieren. Weiterhin haben sie ein Verständnis für die Realisierungsproblematik von eingebettete System entwickelt und kennen die Konzepte und Vorgehensweisen für die Entwicklung eingebetteter Systeme.</p> <p><b>Schlüsselqualifikationen:</b> Erlernen des eigenständigen Arbeitens mit Lehrbüchern (oder englischsprachiger Fachliteratur); Erwerb von Abstraktionsfähigkeiten.</p> <p><b>ACHTUNG:</b> Die Veranstaltung überschneidet sich inhaltlich mit "Softwarearchitekturen und Technologien für eingebettete Systeme". Wer die genannte Veranstaltung bereits gehört hat, kann diese Vorlesung nicht mehr belegen!</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 180 Std. 30 Std. Übung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 45 Std. Vorlesung (Präsenzstudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p>		
<b>Voraussetzungen:</b> Frühere Veranstaltung "Softwarearchitekturen und Technologien für eingebettete Systeme" darf <b>nicht</b> belegt worden sein wegen Überschneidungen.		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 5	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteil</b>
<p><b>Modulteil: Softwarearchitekturen und Enterprise Architecture Management (Vorlesung)</b></p> <p><b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 3</p>
<p><b>Inhalte:</b> Der Vorlesungsinhalt umfasst Patterns, Modellierungstechniken und die Evaluation von Softwarearchitekturen. Weiterhin wird auf den Bereich des Enterprise Architecture Managements eingegangen.</p>
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Bass et al: Software Architecture in Practice</li> <li>• Clements et al: Documenting Software Architectures</li> <li>• Clements et al: Evaluation of Software Architectures</li> <li>• Kopetz: Real-Time Systems</li> </ul>
<p><b>Modulteil: Softwarearchitekturen und Enterprise Architecture Management (Übung)</b></p> <p><b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>



**Prüfung**

**Softwarearchitekturen und Enterprise Architecture Management (mündl. Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

<b>Modul INF-0035: Agile Methoden</b>		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an den Modulveranstaltungen ist der Studierende in der Lage, Agile Methoden für eigene Projekte anzuwenden, zu analysieren und zu bewerten.		
<b>Schlüsselqualifikationen:</b> Erlernen des eigenständigen Arbeitens mit Lehrbüchern (oder englischsprachiger Fachliteratur); Erwerb von Abstraktionsfähigkeiten.		
<b>Arbeitsaufwand:</b> Gesamt: 180 Std. 30 Std. Übung (Präsenzstudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 45 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> Modul Softwaretechnik (INF-0120) - empfohlen		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 5	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Moduleile</b>
<b>Modulteil: Agile Methoden (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 3
<b>Inhalte:</b> Diese Vorlesung vermittelt einen Überblick über aktuelle Methoden wie SCRUM und XP und stellt die Beziehung Agiler Methoden zum Toyota Way her. Der Hauptteil besteht aus Tutorials zur Durchführung eines agil geführten Projektes.
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Folien</li> <li>• The Art of Agile Development Jim Shore, Shane Warden O'Reilly, Beijing u. a. 2008, ISBN 978-0-596-52767-9</li> <li>• Agiles Projektmanagement mit Scrum, Ken Schwaber, Microsoft Press Deutschland, 4. Oktober 2007</li> <li>• Kanban. Evolutionäres Change Management für IT-Organisationen. David J. Anderson</li> </ul>
<b>Modulteil: Agile Methoden (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2

<b>Prüfung</b> <b>Agile Methoden (Klausur)</b> Klausur / Prüfungsdauer: 60 Minuten
--

<b>Modul INF-0036: Praktikum Business &amp; Information Systems Engineering IV (MA)</b>		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an den Modulveranstaltungen ist der Studierende in der Lage weiterführende Techniken im Bereich Business und Information Systems verstehen, anzuwenden und zu bewerten.		
<b>Schlüsselqualifikationen:</b> Erlernen des eigenständigen Arbeitens mit Lehrbüchern (oder englischsprachiger Fachliteratur); Erwerb von Abstraktionsfähigkeiten		
<b>Arbeitsaufwand:</b> Gesamt: 180 Std. 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 90 Std. Praktikum (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Praktikum Business &amp; Information Systems Engineering IV</b>		
<b>Lehrformen:</b> Praktikum		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 6		
<b>Inhalte:</b> Der Schwerpunkt liegt auf interessanten Themen aus dem Bereich wertorientiertes Prozess- und Kundenmanagement		
<b>Literatur:</b> abhängig vom Thema		
<b>Prüfung</b>		
<b>Praktikum Business &amp; Information Systems Engineering IV</b> Mündliche Prüfung / Prüfungsdauer: 30 Minuten		

<b>Modul INF-0037: Praktikum Automotive Software Engineering</b>		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an den Modulveranstaltungen ist der Studierende in der Lage die Grundlagen des Automotive Software Engineerings zu verstehen, anzuwenden und zu bewerten.		
<b>Schlüsselqualifikationen:</b> Erlernen des eigenständigen Arbeitens mit Lehrbüchern (oder englischsprachiger Fachliteratur); Erwerb von Abstraktionsfähigkeiten		
<b>Arbeitsaufwand:</b> Gesamt: 180 Std. 90 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
<b>Voraussetzungen:</b> Empfohlen wird die Teilnahme an einem der beiden links aufgeführten Seminare.  Modul Seminar Grundlagen des Software Engineering für Automotive Systems (BA) (INF-0027) - empfohlen Modul Seminar Grundlagen des Software Engineering für Automotive Systems (MA) (INF-0040) - empfohlen		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: <a href="#">Praktikum Automotive Software Engineering</a></b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 6		
<b>Inhalte:</b> Im Automotive-Praktikum lernen die Teilnehmer wie verschiedene ausgewählte Funktionen innerhalb von Fahrzeugen simuliert und analysiert werden können. In einem zweitägigen Einführungskurs werden die benötigten theoretischen Grundlagen für die Bearbeitung der Praxisaufgabe gelegt. Während des Einführungskurses wird das Modell eines Antiblockiersystems (ABS) auf realen Steuergeräten behandelt. Die Teilnehmer lernen dabei u.a. die im Automotive-Umfeld häufig eingesetzte Modellierungswerkzeug-Kombination „Matlab/Simulink“ sowie das graphische Simulations- und Analyse-Tool „CarMaker“ kennen und erhalten einen praktischen Einblick in die Funktionsweise von FlexRay-Bussystemen. Nach dem Einführungskurs soll in Gruppen von zwei bis drei Teilnehmern das Modell eines ACC-Systems (Adaptive Cruise Control) erstellt, simuliert und verifiziert werden.		
<b>Literatur:</b> abhängig vom Thema		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Praktikum zu Automotive Software Engineering</b> (Praktikum) Im Automotive-Praktikum lernen die Teilnehmer/innen wie Software für Automotive-Anwendungen entwickelt, simuliert und analysiert werden kann. In einem Einführungskurs werden wir uns die notwendigen Grundlagen anhand von eigens dafür konzipierten Tutorials erarbeiten. Die Teilnehmer/innen lernen dabei u.a. die im Automotive-Umfeld häufig eingesetzte Modellierungswerkzeug-Kombination „Matlab/Simulink“ sowie das		

graphische Simulations- und Analyse-Tool „CarMaker“ kennen und erhalten einen praktischen Einblick in die Funktionsweise von FlexRay-Bussystemen. Für die erstellen Modelle kann mit der vorhandenen Toolchain automatisiert C-Code erzeugt werden. Dieser kann auf realen Steuergeräten simuliert, getestet und analysiert werden. Im Anschluss an den Einführungskurs werden in Gruppen von zwei bis drei Teilnehmern die Übungsaufgaben bearbeitet, darin geht es u.a. um die Modelle eines ABS (Anti-Blockier-System) und eines ACC (Adaptive Cruise Control). Im Abschlussprojekt modellieren, impleme  
... (weiter siehe Digicampus)

**Prüfung**

**Praktikum Automotive Software Engineering**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

<b>Modul INF-0038: Praktikum Avionic Software Engineering</b>		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an den Modulveranstaltungen ist der Studierende in der Lage die Grundlagen des Avionic Software Engineerings zu verstehen, anzuwenden und zu bewerten.		
<b>Schlüsselqualifikationen:</b> Erlernen des eigenständigen Arbeitens mit Lehrbüchern (oder englischsprachiger Fachliteratur); Erwerb von Abstraktionsfähigkeiten		
<b>Arbeitsaufwand:</b> Gesamt: 180 Std. 90 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
<b>Voraussetzungen:</b> Empfohlen wird die Teilnahme an einem der beiden Seminare.  Modul Seminar Grundlagen des Software Engineering für Avionic Systems (BA) (INF-0028) - empfohlen Modul Seminar Grundlagen des Software Engineering für Avionic Systems (MA) (INF-0041) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<b>Moduleil: Praktikum Avionic Software Engineering</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 6		

**Inhalte:**

Die Teilnehmer im Avionik-Praktikum erlernen, wie Software für komplexe Avionik-Systeme entwickelt wird. In kleinen Gruppen soll von den Studenten ein einfacher Autopilot für ein fliegendes System umgesetzt und in einer Simulationsumgebung getestet werden.

Die Studenten erhalten hierzu eine Spezifikation der zu implementierenden Funktionen, sowie ein Framework zur Anbindung des zu entwickelnden Autopilots an eine Simulationsumgebung (X-Plane).

In einer Einführungs-Blockveranstaltung erwerben die Teilnehmer die nötigen Grundkenntnisse über die Entwicklung zuverlässiger Avionik-Systeme und erhalten einen Überblick über die für dieses Praktikum verwendeten Technologien:

- Techniken zur Entwicklung sicherheitskritischer Systeme
- Relevante Standards und rechtliche Rahmenbedingungen in der Luft- und Raumfahrt
- Qualitätssicherung in der Software-Entwicklung durch den Einsatz geeigneter Werkzeuge
- Einführung in die Steuerung fliegender Systeme und Navigation
- Komponentenbasierte Software-Entwicklung mit Java und OSGi
- Echtzeitfähige Software in Java gemäß der RTSJ-Spezifikation

Das Praktikum wird in den Semesterferien angeboten und besteht aus dem theoretischen Teil als Blockveranstaltung und der anschließenden selbstständigen Umsetzung der Praktikumsaufgabe durch die Studenten.

**Die erforderlichen Tätigkeiten sind:**

- Erstellung einer geeigneten Software-Architektur und -Design
- Implementierung eines grundlegenden Autopilots innerhalb des vorgebenen Frameworks in Java und OSGi
- Überprüfung der funktionalen Korrektheit durch Unit- und Integrationstests

**Vorkenntnisse:**

- Grundkenntnisse im Bereich Software Engineering
- Programmiererfahrung in Java
- Interesse an Avionik-Systemen
- **Keine** Erfahrung mit OSGi erforderlich!

**Literatur:**

abhängig vom Thema

**Zugeordnete Lehrveranstaltungen:**

**Praktikum Avionic Software Engineering** (Praktikum)

Das Praktikum Avionic Software Engineering vermittelt den Teilnehmern alle nötigen Sprachen, Werkzeuge und Vorgehensweise zur Implementierung eines Autopilots wie er in handelsüblichen UAVs gefunden werden kann.

**Prüfung**

**Praktikum Avionic Software Engineering**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

<b>Modul INF-0039: Seminar über Software Engineering verteilter Systeme (MA)</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet des Software Engineerings verteilter Systeme selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken; Abwägen von Lösungsansätzen</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Seminar über Software Engineering verteilter Systeme</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>		
<b>Inhalte:</b> Aktuelle Software Engineering-Themen aus Industrie und Forschung.		
<b>Literatur:</b> Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.		
<p><b>Zugeordnete Lehrveranstaltungen:</b> <b>Seminar zu Software Engineering verteilter Systeme f. Master (Seminar)</b> Bestandteil dieses Seminars sind fortgeschrittene Ansätze und Techniken im Bereich Software Engineering. Dies betrifft alle Phasen des Softwareentwicklungszyklus von der Anforderungsanalyse bis hin zum Testen. Modellierungstechniken sowie domänenspezifische Sprachen bilden einen Schwerpunkt des Seminars. Unter anderem werden in diesem Seminar Themen in Kooperation mit dem Kernkompetenzzentrum FIM vergeben.</p>		
<p><b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar</p>		



<b>Modul INF-0040: Seminar Grundlagen des Software Engineering für Automotive Systems (MA)</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet des Automotive Software Engineerings selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken; Abwägen von Lösungsansätzen</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Seminar Grundlagen des Software Engineering für Automotive Systems</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>		
<p><b>Inhalte:</b> Diese Seminar soll die Grundlagen des Systems &amp; Software Engineering im Automotive Bereich behandeln. Es werden dabei Aspekte der Vorlesung Automotive Software Engineering aufgenommen und vertieft.</p>		
<p><b>Literatur:</b> Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.</p>		
<p><b>Zugeordnete Lehrveranstaltungen:</b> <b>Seminar zu Automotive Software Engineering (Master) (Seminar)</b> Bestandteil dieses Seminars sind Ansätze und Techniken im Bereich des Automotive Software Engineering. Dies betrifft alle Phasen des System- und Softwareentwicklungszyklus von der Anforderungsanalyse bis hin zum Testen. Themen behandeln verschieden Aspekte daraus, u.a. Modellierungstechniken, domänenspezifische Sprachen, autonomes Fahren und Problemstellung durch den Einsatz von Multicore-Systemen.</p>		
<p><b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar</p>		

<b>Modul INF-0041: Seminar Grundlagen des Software Engineering für Avionic Systems (MA)</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet des Avionic Software Engineerings selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.		
<b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken; Abwägen von Lösungsansätzen		
<b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Seminar Grundlagen des Software Engineering für Avionic Systems</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>SWS:</b> 2
<b>Inhalte:</b> Dieses Seminar soll die Grundlagen des Systems & Software Engineering im Avionic Bereich behandeln. Es sind verschiedene Themen zu bearbeiten die als Grundlage für ein nachfolgendes Praktikum dienen sollen.
<b>Literatur:</b> Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Seminar zu Avionic Software Engineering (Master) (Seminar)</b> Bestandteil dieses Seminars sind Ansätze und Techniken im Bereich des Avionic Software Engineering. Dies betrifft alle Phasen des System- und Softwareentwicklungszyklus von der Anforderungsanalyse bis hin zum Testen. Themen behandeln verschieden Aspekte daraus, u.a. Modellierungstechniken, domänenspezifische Sprachen und autonomes Fliegen.

<b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar
---

<b>Modul INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet des Software Engineerings verteilter Systeme zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Grundsätze guter wissenschaftlicher Praxis; Teamfähigkeit; Erlernen von Präsentationstechniken; schriftliche Präsentation eigener Ergebnisse</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 0	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Projektmodul Softwaremethodiken für verteilte Systeme</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 1</p>		
<b>Inhalte:</b> Aktuelle Forschungsthemen am DS-Lab.		
<b>Literatur:</b> Wird zu den jeweiligen Themen bereitgestellt.		
<p><b>Prüfung</b> <b>Projektabnahme, Vortrag, Abschlussbericht</b> Praktikum</p>		

<b>Modul INF-0066: Organic Computing II</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p><b>Lernziele/Kompetenzen:</b> Fundierte Kenntnisse über das Forschungsgebiet Organic Computing und die Funktionsweise selbstorganisierender Systeme. Verständnis für Probleme beim Entwurf von komplexen vernetzten Systemen und von forschungsorientierten Lösungsansätzen.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Organic Computing II (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>		
<p><b>Inhalte:</b> Die Vorlesung "Organic Computing" vermittelt Ansätze zur Organisation von komplexen vernetzten Systemen, die aus einer Vielzahl von autonomen Teilsystemen bestehen. Dazu werden zunächst Anforderungen und Ziele solcher Systeme definiert und diskutiert. Darüber hinaus werden Konzepte aus dem Bereich der Systemarchitekturen und Ansätze aus dem Bereich naturalogischer Algorithmen dargestellt und bewertet. In allen Teilen werden Bezüge zu konkreten Anwendungsgebieten gegeben. Die zugehörige Übung bietet die Möglichkeit, die erlernten Ansätze zu vertiefen und beispielhaft anzuwenden.</p>		
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Folien</li> <li>• Müller-Schloer et al.: Organic Computing - A Paradigm Shift for Complex Systems, Birkhäuser Verlag, Basel, 2011, ISBN 978-3034801294</li> <li>• Würtz (ed.): Organic Computing (Understanding Complex Systems), SpringerVerlag Berlin, 2008, ISBN 978-3540776567</li> <li>• Mitchell: Machine Learning, The McGraw-Hill Companies, 1997, ISBN 978-0071154673</li> <li>• Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989, ISBN 978-0201157673</li> <li>• Michalewicz, Fogel: How to Solve it: Modern Heuristics, Springer Verlag Berlin, 2004, ISBN 978-3540224945</li> <li>• Tomforde: Runtime Adaptation of Technical Systems, Südwestdeutscher Verlag für Hochschulschriften, 2012, ISBN 978-3838131337</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b>		

**Organic Computing II (Vorlesung)** (Vorlesung)

Organic Computing ist ein neues Paradigma des Systems Engineering technischer Systeme die in der "echten Welt" realisiert werden. Aspekte wie Anpassungsfähigkeit an sich ständig ändernde Bedingungen der technischen Umgebung sowie die Implementierung sog. Self-X Eigenschaften gehören zu den Herausforderungen, welche Organic Computing fokussiert. In der Vorlesung werden Konzepte und Methoden diskutiert, die den Entwurf und die Realisierung von Organic Computing Systemen erlauben.

**Modulteil: Organic Computing II (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Zugeordnete Lehrveranstaltungen:**

**Organic Computing II (Übung)** (Übung)

**Prüfung**

**Organic Computing II (mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

**Beschreibung:**

Die Prüfung kann jedes Semester zu Beginn und Ende der vorlesungsfreien Zeit abgelegt werden.

<b>Modul INF-0068: Interactive Simulation</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p><b>Lernziele/Kompetenzen:</b> In this course, the students are taught foundational knowledge about interactive simulations. In particular, in-depth apprehension of methods in the fields of modelling &amp; simulation, representation, numerical methods and computer graphics will empower the student to evaluate and to contribute to the design and the programmatic implementation of interactive simulations.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> letzte Prüfung: SS 2017	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<b>Modulteil: Interactive Simulation (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Englisch <b>SWS:</b> 2		
<p><b>Inhalte:</b> The basic concept of modelling &amp; simulation is extended by the notion of user interactions. Differences and common features among several academic and industrial examples will be stressed in order to develop a generalised terminology and methodology for interactive simulations. Interactivity translates into one or several users influencing the model and the simulation process, respectively; accordingly, the course revolves around the changes to the simulation model and the emerging dynamics in respect to the computational processes that result from the introduction of user interactions. Interactivity in simulations necessitates the development and the utilisation of real-time rendering techniques (computer graphics), intense efforts towards optimisation, and a clear understanding of acceptable numerical errors due to systematic approximations. In this course, we shed light on the state-of-the-art and discuss current challenges and their potential solutions, for instance in regard to simulation histories or dynamic abstraction.</p>		
<b>Literatur:</b> aktuelle wissenschaftliche Paper		
<b>Modulteil: Interactive Simulation (Übung)</b>		
<b>Lehrformen:</b> Übung <b>Sprache:</b> Englisch <b>SWS:</b> 2		

**Prüfung**

**combined exam: written project report ...**

Hausarbeit/Seminararbeit

**Prüfung**

**... and oral 15 min (combined exam)**

Mündliche Prüfung / Prüfungsdauer: 15 Minuten

**Beschreibung:**

Die Prüfung kann jedes Semester zu Beginn und Ende der vorlesungsfreien Zeit abgelegt werden.

<b>Modul INF-0070: Seminar Organic Computing</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage zur selbstständigen Erarbeitung eines Themas und der geeigneten Präsentation in Schrift und Vortrag, sowie der sachlichen Diskussion über einen Vortrag.		
<b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis		
<b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Organic Computing</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> Die Themen des Seminars werden jedes Jahr neu festgelegt und aktuellen Trends angepasst.		
<b>Literatur:</b> Literatur in Abhängigkeit von den aktuellen Themen: wiss. Paper oder Bücher		
<b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar		



<b>Modul INF-0071: Seminar Naturalloge Algorithmen und Multiagentensysteme</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, spezifische Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien im Schnittbereich naturalogener Verfahren und Multiagentensysteme selbstständig zu erarbeiten und bezogen auf ein spezielles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die Arbeitstechniken, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> selbstständiges wissenschaftliches Arbeiten</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Naturalloge Algorithmen und Multiagentensysteme</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<p><b>Inhalte:</b> In dem Seminar sollen sich die Studenten jeweils einem speziellen Thema in Schnittbereich naturaloge Algorithmen und Multiagentensysteme genauer beschäftigen. Diese Thema kann ein bestimmte Anwendung, z.B. das Swarmoid-Projekt, sein oder auch eine bestimmte Technik, z.B. für Task Allocation betreffen. Sie erstellen einen etwa 30-minütigen Vortrag zum gegebenen individuellen Thema. In einer schriftlichen Ausarbeitung werden die Erkenntnisse zum Thema zusammengefasst.</p>		
<p><b>Literatur:</b> wird im Seminar bekanntgegeben</p>		
<p><b>Zugeordnete Lehrveranstaltungen:</b> <b>Naturalloge Algorithmen und Multi-Agenten Systeme (Seminar)</b> (Seminar) Es handelt sich um eine Master-Veranstaltung. Es werden max. 10 Themen vergeben. Die Zuordnung eines Themas erfolgt per Lösung, aber unter Berücksichtigung von Präferenzen.</p>		
<b>Prüfung</b>		
<b>Vortrag und schriftliche Ausarbeitung</b>		
Seminar		

<b>Modul INF-0072: Projektmodul Organic Computing</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet "Organic Computing" zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Grundsätze guter wissenschaftlicher Praxis, selbstständiges Arbeiten, Erlernen des Arbeitens mit englischsprachiger Fachliteratur, analytisch-methodische Kompetenz</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<p><b>Modulteil: Projektmodul Organic Computing</b>  <b>Lehrformen:</b> Praktikum  <b>Sprache:</b> Deutsch  <b>SWS:</b> 1</p>
<p><b>Inhalte:</b> Mitarbeit an aktuellen Forschungsthemen.</p>
<p><b>Literatur:</b> In Abhängigkeit vom zu bearbeitenden Thema:</p> <ul style="list-style-type: none"> <li>• Paper</li> <li>• Buch</li> <li>• Handbuch</li> </ul>

<p><b>Prüfung</b>  <b>Vortrag und Abschlussbericht</b>  Praktikum</p>
---

<b>Modul INF-0108: Projektmodul Lehrprofessur für Informatik</b> <i>Project Module Teaching Professorship Informatics</i>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Robert Lorenz		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul haben die Studierenden tiefergehende Fachkenntnisse und Fähigkeiten auf einem der Gebiete "Nebenläufige Systeme" und "Semantische Dialogmodellierung" erworben, die es ihnen ermöglichen, an die internationale Forschung anzuknüpfen. Sie sind fähig, innovative Methoden bei der Lösung von Problemen in diesem Gebiet anzuwenden und einen wissenschaftlichen Beitrag zu diesem Gebiet zu leisten.</p> <p>Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Recherche in englischsprachiger Literatur; Verständliche, sichere und überzeugende Präsentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewußtsein; Kommunikationsfähigkeit; Fertigkeit der Zusammenarbeit in Teams und Verstehen von Teamprozessen; Grundsätze guter wissenschaftlicher Praxis; Projektmanagementfähigkeiten; Wissenschaftliche Methodik;</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
<b>Voraussetzungen:</b> Grundkenntnisse in einschlägigen Forschungsgebieten des Lehrstuhls		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<p><b>Modulteil: Projektmodul Lehrprofessur für Informatik</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch / Englisch <b>SWS:</b> 1</p>		
<p><b>Inhalte:</b> Mitarbeit an der Entwicklung formaler Grundlagen und theoretischer Ergebnisse, dem Entwurf und der Programmierung unterstützender Softwaretools und der Evaluation von Ergebnissen und Konzepten in aktuellen Forschungsprojekten des Lehrstuhls aus den Bereichen "Nebenläufige Systeme" und "Semantische Dialogmodellierung". Mögliche Themen: Synthese von Petrinetzen aus nicht-sequentiellen Verhaltensbeschreibungen, Process Mining Techniken, Entfaltung von Petrinetzen und Entfaltungs-basiertes Model-Checking, Finite State Transducer in der semantischen Dialogmodellierung, Petrinetz-Transduktoren, Hierarchische kognitive dynamische Systeme zur Signalverarbeitung, Dialog-Strategien, Konfiguration von Spracherkennern, Benutzermodelle in der Spracherkennung, Wizard-of-Oz Experimente zur Erstellung lokaler Grammatiken, Unifikationsalgorithmen</p>		

**Literatur:**

- J. Desel, W. Reisig, G. Rozenberg: Lectures on Concurrency and Petri Nets, Springer, Lecture Notes in Computer Science 3098, 2004
- Projekt-Homepage VipTool: [http://www.fernuni-hagen.de/sttp/forschung/vip\\_tool.shtml](http://www.fernuni-hagen.de/sttp/forschung/vip_tool.shtml)
- Projekt-Homepage SYNOPS: <http://www.informatik.uni-augsburg.de/lehrstuehle/inf/projekte/synops/>
- Daniel Jurafsky & James H. Martin: Speech and Language Processing
- M. Huber; C. Kölbl; R. Lorenz; R. Römer; G. Wirsching: Semantische Dialogmodellierung mit gewichteten Merkmal-Werte-Relationen. In: Rüdiger Hoffmann (Hrsg.), Elektronische Sprach-signalverarbeitung 2009, Tagungsband der 20. Konferenz, 2009, Studentexte zur Sprachkommunikation 54, Seiten 25-32
- M. Droste, W. Kuich, H. Vogler (Eds.): Handbook of Weighted Automata. Monographs in Theoretical Computer Science, Springer, 2009.
- A. Esposito (Eds.): Behavioral Cognitive Systems. LNCS 7403, Springer, 2012

**Prüfung**

**Vortrag und Abschlussbericht**

Praktikum

<b>Modul INF-0129: Softwaretechnik II</b>		8 ECTS/LP
Version 2.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage, fortgeschrittene Verfahren der agilen Softwareentwicklung, des Requirements Engineerings, des Testens und des Refactorings von Softwaresystemen anzuwenden. Sie sind in der Lage, die Eignung verschiedener Softwareentwicklungsprozesse für konkrete Projekte zu bewerten. Sie sind in der Lage, wesentliche Methoden der Requirements-Erfassung und Dokumentation anzuwenden und die Eignung verschiedener Dokumentationsformen zu bewerten. Sie können systematisch Kundenanforderungen analysieren. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken und können geeignete Methoden und Entwurfsalternativen auswählen und anwenden. Sie kennen Verfahren zur Bestimmung von Testfällen und zur Bewertung der Adäquatheit von Testsuiten und können diese anwenden. Sie kennen die Prinzipien guten objektorientierten Designs und können durch Refactoring die Qualität des Designs eines Systems verbessern. Sie können Ideen und Konzepte sicher und überzeugend darstellen und haben die Fähigkeit zur Zusammenarbeit im Team.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Moderieren fachlicher Sitzungen, Erlernen des eigenständigen Arbeitens mit Lehrbüchern und Arbeit in selbstorganisierten Teams, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)</p>		
<p><b>Voraussetzungen:</b> Programmierkenntnisse in Java (empfohlen) Modul Softwaretechnik (INF-0120) - empfohlen</p>		
<p><b>Angebotshäufigkeit:</b> jedes Wintersemester</p>	<p><b>Empfohlenes Fachsemester:</b> ab dem 1.</p>	<p><b>Minimale Dauer des Moduls:</b> 1 Semester</p>
<p><b>SWS:</b> 6</p>	<p><b>Wiederholbarkeit:</b> siehe PO des Studiengangs</p>	
<p><b>Modulteile</b></p>		
<p><b>Modulteil: Softwaretechnik II (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>		

**Inhalte:**

**Agile Softwareentwicklung:**

- Entwicklungsmethoden (Scrum)
- Agile Praktiken
- Agile Werte, Prinzipien und Methoden

**Refactoring**

- Code Smells
- Prinzipien des objektorientierten Designs
- Wichtige Refactorings

**Testen**

- Testprozess und Ziele des Testens
- Testarten
- Methoden zur Testfallgewinnung
- Adäquatheitskriterien beim Blackbox- und Whitebox-Testen

**Requirements Engineering**

- Aufgaben, Begriffe und Artefakte
- RE-Prozess
- Techniken zur Requirements-Elicitation, -Analyse und -Dokumentation
- Qualitätskriterien für Software-Requirements

**Literatur:**

- Pohl, Rupp: Basiswissen Requirements Engineering, dpunkt Verlag 2009
- U. Hammerschall, G. Benekean: Software Requirements, Pearson 2013
- S. Robertson, J. Robertson: Mastering the Requirements Process, Addison-Wesley 2013
- Bleek, Wolf: Agile Softwareentwicklung, dpunkt Verlag 2008
- R. Pichler: Scrum, dpunkt.verlag 2008
- Spillner, Linz: Basiswissen Softwaretest, dpunkt Verlag 2005
- Fowler: Refactoring, Addison-Wesley 1999
- Vorlesungsfolien mit schriftlichen Ergänzungen und Anmerkungen

**Modulteil: Softwaretechnik II (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 4

**Prüfung**

**Softwaretechnik II Klausur**

Klausur / Prüfungsdauer: 90 Minuten

<b>Modul INF-0130: Formale Methoden im Software Engineering</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<b>Lernziele/Kompetenzen:</b> Die Studierenden können formale Methoden für die Programmverifikation, speziell bei sicherheitskritischer Software einsetzen. Sie trainieren die Fertigkeit zum logischen und analytischen Denken. Sie können Spezifikationen von Datenstrukturen erstellen und deren Eigenschaften formal beweisen. Sie sind in der Lage, funktionale Eigenschaften von Programmen zu formulieren und dafür Beweise zu entwickeln. Sie haben die Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen und können mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen abgeben.		
<b>Schlüsselqualifikationen:</b> Training des logischen Denkens, analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Formale Methoden im Software Engineering (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> Übergeordnetes Ziel ist die Produktion beweisbar korrekter Software. In der Vorlesung werden verschiedene klassische Methoden für die Programmverifikation im Kleinen behandelt. Darüber hinaus werde innovative Techniken für die formale Modellierung und Verifikation großer Systeme vermittelt. Als Werkzeug kommt das KIV-System zum Einsatz, das die formale Spezifikation und Verifikation von Systemen ermöglicht. Konkrete Inhalte sind: Algebraische Spezifikationen, interaktives Theorembeweisen, Hoare-Logik, Dynamische Logik, Temporallogik		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Sperschneider, Antoniou: Logic: A Foundation for Computer Science, Addison Wesley 1991</li> <li>• Loeckx, Ehrich, Wolf: Specification of Abstract Data Types, Wiley 1996</li> <li>• Ausführliche Dokumentation</li> <li>• Folienhandout</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b>		
<b>Formale Methoden im Software Engineering (Vorlesung)</b> Die Verwendung formaler Methoden bei der Entwicklung korrekter Software steht an der Schwelle der kommerziellen Nutzung. Das KIV-System ist ein Werkzeug, das die formale Spezifikation, Verifikation und		

Synthese von Programmen ermöglicht. Es wird seit mehreren Jahren entwickelt und inzwischen in industriellen Studien erprobt. Übergeordnetes Ziel ist die Produktion beweisbar korrekter Software. Die Lehrveranstaltung vermittelt den "state of the Art" des Einsatzes formaler Methoden bei der Softwareentwicklung. Es werden Spezifikationstechniken zur Beschreibung und Methoden zum Nachweis der Korrektheit von Softwaresystemen behandelt. Die Lehrveranstaltung beginnt mit der (algebraischen) Spezifikation von Datentypen und stellt dann Kalküle und Vorgehensweisen für die Verifikation sequentieller und paralleler Programme vor. Weiterhin wird Refinement als systematisches Entwicklungsvorgehen von einer abstrakten Spezifikation zu einer korrekten Implementierung vorgestellt. Die Rechner im Raum  
... (weiter siehe Digicampus)

**Modulteil: Formale Methoden im Software Engineering (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 4

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Formale Methoden im Software Engineering (Übung)**

**Prüfung**

**Formale Methoden im Software Engineering (mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 45 Minuten



<b>Modul INF-0131: Software- und Systemsicherheit</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden können Bedrohungsanalyse durchführen, kryptographische Protokolle entwickeln, Chipkarten programmieren und sicherheitskritische Systeme entwerfen. Sie können systematisch Bedrohungen für Softwaresysteme analysieren und deren Risiken bewerten. Sie können einen modellgetriebenen Entwicklungsprozess für sicherheitskritische Systeme anwenden. Sie sind in der Lage, fachliche Lösungskonzepte in Programme umzusetzen. Sie haben Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<p><b>Modulteil: Software- und Systemsicherheit (Vorlesung)</b>  <b>Lehrformen:</b> Vorlesung  <b>Sprache:</b> Deutsch  <b>SWS:</b> 2</p>
<p><b>Inhalte:</b>          Inhalt der Vorlesung ist der Entwurf sicherer Softwaresysteme, speziell verteilter Systeme, deren Sicherheit wesentlich auf dem Einsatz von Sicherheitsprotokollen beruht. Ein Schwerpunkt liegt dabei auf Anwendungen, in denen Chipkarten eingesetzt werden. In der Vorlesung werden Kenntnisse in JavaCard, der Chipkartentechnologie, Bedrohungsanalyse und dem Design kryptographischer Anwendungsprotokolle vermittelt, die in den Übungen an praktischen Beispielen (u.a. einer elektronischen Kopierkarte und einer elektronischen Fahrkarte) erprobt werden. Bei der Entwicklung der Protokolle wird der SecureMDD-Ansatz verwendet, eine Methode zur modellgetriebenen Entwicklung sicherheitskritischer Protokolle.</p>
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Schneier: Applied Cryptography, Wiley and Sons, 1996 (2nd edition)</li> <li>• Anderson, Needham: Programming Satan's Computer, in: Computer Science Today, Springer LNCS 1000, 1995</li> <li>• Lowe: Breaking and fixing the Needham-Schroeder public-key protocol using FDR, in: Tools and Algorithms for the Construction and Analysis of Systems, Springer LNCS 1055, 1996</li> <li>• Folienhandout, Spezifikationen und APIs</li> </ul>

**Modulteil: Software- und Systemsicherheit (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 4

**Prüfung**

**Software- und Systemsicherheit (mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

<b>Modul INF-0133: Selbstorganisierende, adaptive Systeme</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden kennen die Eigenschaften und den Aufbau selbst-organisierender Systeme aus der Biologie, Soziologie, Physik und anderen Bereichen und der systematischen Modellierung und Konstruktion adaptiver Systeme in der Informatik und können solche Systeme analysieren und selbst entwerfen. Sie kennen Vor- und Nachteile verschiedener Entwurfsalternativen und können sie im Kontext der Problemstellung bewerten. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken und können geeignete Methoden auswählen und anwenden und wissenschaftlich aussagekräftige Bewertungen abgeben.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Selbstorganisierende, adaptive Systeme (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<p><b>Inhalte:</b> In der Vorlesung werden die Grundlagen verschiedener Selbst-Organisationsmechanismen sowie das Handwerkszeug, um diese in IT-Systemen einsetzen zu können, vermittelt. Im Verlauf der Veranstaltung werden verschiedene Beispiele für selbstorganisierende Systeme vorgestellt, untersucht und Anwendungen der erlernten Organisationsprinzipien auf Beispiele aus der Informatik erläutert. Schließlich werden Methoden betrachtet, mit deren Hilfe sich Selbst-Organisation und Adaptivität in die Entwicklung komplexer Computersysteme integrieren lassen. Konkrete Themen sind: Selbst-Organisation, Emergenz, Chaostheorie, zelluläre Automaten, Spieltheorie, Multi-Agentensysteme, Autonomic Computing, Organic Computing.</p>		
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Gleick: Chaos: Making a New Science, Penguin 2008</li> <li>• Strogatz: Sync : The Emerging Science of Spontaneous Order, Hyperion 2003</li> <li>• Miller, Page: Complex Adaptive Systems: An Introduction to Computational Models of Social Life, Princeton University Press 2007</li> <li>• Dawkins: The Selfish Gene, Oxford University Press, 3rd Revised Edition</li> <li>• Wolfram: A New Kind of Science, Wolfram Media Inc. 2002</li> <li>• von Neumann, Morgenstern: Theory of Games and Economic Behavior, 2004</li> <li>• Folienhandout</li> </ul>		

---

**Modulteil: Selbstorganisierende, adaptive Systeme (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 4

**Prüfung**

**Selbstorganisierende, adaptive Systeme (mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

<b>Modul INF-0136: Seminar Software- und Systems Engineering (Master)</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage selbstständig ein wissenschaftlich anspruchsvolles Thema aus dem Gebiet der Softwaretechnik zu erarbeiten, geeignet in Schrift und Vortrag zu präsentieren und sachlich über Vorträge zu diskutieren. <b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis		
<b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Software- und Systems Engineering (Master)</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>Angebotshäufigkeit:</b> jedes Sommersemester <b>SWS:</b> 2		
<b>Inhalte:</b> Die konkreten Themen des Seminars beschäftigen sich mit aktuellen Themen des Software- und Systems Engineering auf Masterniveau und werden jedes Jahr neu festgelegt und an neue Entwicklungen angepasst.		
<b>Literatur:</b> abhängig von den konkreten Themen des Seminars		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Seminar zu Software- und Systems Engineering (Master) (Seminar)</b> In dem Seminar werden aktuelle Themen aus dem Umfeld des Software- und Systems Engineering behandelt. Es stehen folgende Themen noch zur Auswahl: * Data-Race Detection am Beispiel RV-Predict ( <a href="https://runtimeverification.com/predict/">https://runtimeverification.com/predict/</a> ) oder JavaPathfinder ( <a href="http://babelfish.arc.nasa.gov/trac/jpf">http://babelfish.arc.nasa.gov/trac/jpf</a> ) * Dynamic Frames zur Verifikation Heap-basierter Programme (siehe "Dynamic Frames: Support for Framing, Dependencies and Sharing Without Restrictions" von Ioannis T. Kassios) * Static Driver Verifier von Microsoft (siehe z.B. <a href="https://www.microsoft.com/en-us/research/project/slam/">https://www.microsoft.com/en-us/research/project/slam/</a> und "SLAM and Static Driver Verifier: Technology Transfer of Formal Methods inside Microsoft" von Thomas Ball, Byron Cook, Vladimir Levin und Sriram K. Rajamani) * eigener Themenvorschlag aus dem Bereich der Nebenläufigkeit, formalen Methoden oder statischen Analyse sind ebenfalls herzlich willkommen bereits vergeben sind: * Software Transactional Memory am Beispiel von ScalaSTM ( <a href="https://nbronson.github.io/scala-stm/">https://nbronson.github.io/scala-stm/</a> ) * ... (weiter siehe Digicampus)		

**Prüfung**

**Vortrag und schriftliche Ausarbeitung**

Seminar

<b>Modul INF-0137: Projektmodul Software- und Systems Engineering</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden haben tiefere Fachkenntnisse und Fähigkeiten auf dem Gebiet der Softwaretechnik erworben, die es ihnen ermöglichen, an die internationale Forschung anzuknüpfen. Sie sind fähig, innovative Methoden bei der Lösung von Problemen in diesem Gebiet anzuwenden und einen wissenschaftlichen Beitrag zu diesem Gebiet zu leisten.</p> <p><b>Schlüsselqualifikationen:</b> Grundsätze guter wissenschaftlicher Praxis, selbstständiges Arbeiten, Erlernen des Arbeitens mit englischsprachiger Fachliteratur, analytisch-methodische Kompetenz</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Projektmodul Software- und Systems Engineering</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 1</p>		
<b>Inhalte:</b> Mitarbeit an aktuellen Forschungsthemen des Lehrstuhls		
<b>Literatur:</b> abhängig von dem konkreten Projekt: wissenschaftliche Papiere, Dokumentation		
<p><b>Prüfung</b> <b>Projektabschluss</b> Praktikum</p>		

<b>Modul INF-0177: Einführung in die Künstliche Intelligenz</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<b>Lernziele/Kompetenzen:</b> Die Studierenden beherrschen basale theoretische und praktische Konzepte aus dem Bereich der künstlichen Intelligenz. Weiterhin sind sie nach Vorlesungsteilnahme in der Lage intelligente Verfahren zu nutzen, zu entwickeln und dabei dem Problem adäquate Methoden einzusetzen.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b>		
keine		
<b>Angebotshäufigkeit:</b> wird nicht mehr angeboten!	<b>Empfohlenes Fachsemester:</b> 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Einführung in die Künstliche Intelligenz (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2
<b>Inhalte:</b> Einführung, Problemlösen mit Suche und Constraint Satisfaction, Wissensrepräsentation und Reasoning, Räumliches und Zeitliches Schliessen, Planen, Reasoning und Planen mit Unsicherheit, Intelligente Anwendungen.
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• S. Russell&amp;P. Norvig: Artificial Intelligence - A Modern Approach, 3rd Edition, 2010</li> <li>• weitere Literatur wird in der Vorlesung angegeben.</li> </ul>

<b>Modulteil: Einführung in die Künstliche Intelligenz (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2
--

<b>Prüfung</b> <b>Klausur Einführung in die Künstliche Intelligenz</b> Klausur
--



<b>Modul INF-0189: Qualitätssicherung im Software Engineering</b>		5 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p><b>Lernziele/Kompetenzen:</b> Nach erfolgreicher Absolvierung der Lehrveranstaltung verstehen Studierende die Grundtechniken der Qualitätssicherung. Sie sind in der Lage, SW-Module zu spezifizieren und kennen die wesentlichen Testtechniken und deren Einsatzzwecke im Software Engineering. Die Studierenden sind für das Thema Qualität im Software Engineering sensibilisiert und können verschiedene Qualitätskriterien/-metriken kritisch hinterfragen und bewerten. Des Weiteren kennen und verstehen sie die Prinzipien von konstruktiven Qualitätssicherungstechniken und -praktiken.</p> <p><b>Schlüsselqualifikationen:</b> Training des logischen Denkens, analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis, Verbesserung der eigenen Softwareentwicklungskompetenz</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> unregelmäßig (i. d. R. im SoSe)	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Qualitätssicherung im Software Engineering (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<p><b>Inhalte:</b> Ingenieursdisziplinen kombinieren Design- und Entwicklungsaktivitäten mit Aktivitäten, die vorläufige und endgültige Produkte prüfen, um Mängel zu erkennen und zu beseitigen. Software Engineering ist hierbei keine Ausnahme: Konstruktion hochqualitativer Software bedarf einer sich ergänzenden Kombination von Maßnahmen des Designs und der Prüfung der Software über den gesamten Entwicklungszyklus hinweg. Gerade aufgrund der Durchdringung der Software von immer mehr kritischen Bereichen wie etwa Automotive oder Avionik rücken Maßnahmen zu Qualitätssicherung immer mehr in den Fokus der Aufmerksamkeit im modernen Software Engineering. In dieser Vorlesung werden Konzepte, Techniken und Methoden der Qualitätssicherung im Software Engineering vermittelt. Dies umfasst, u.a., die Spezifikation von Software in einem Kontinuum von natürlichsprachlicher bis formalsprachlicher Notation, automatisierte Methoden und Techniken zur analytischen sowie auch zur konstruktivistischen Qualitätssicherung, Entwicklung von Qualitätssicherungsstrategien sowie Grundlagen im Umgang mit gängigen Werkzeugen, die im Software Engineering zum Einsatz kommen. Den Abschluss bildet die kritische Auseinandersetzung mit formalen Methoden, die für besonders kritische Module zum Einsatz kommen können und in Zertifizierungsstandards anerkannt werden.</p>		

**Literatur:**

- P. Ammann und J. Offutt: Introduction to Software Testing. Cambridge University Press, 2008.
- M. Pezzè und M. Young: Software Testing and Analysis: Process, Principles, and Techniques. Wiley & Sons, 2008.
- R. Binder: Testing Object-Oriented Systems: Models, Patterns, and Tools. Addison-Wesley, 2000.
- M. Chemuturi: Mastering Software Quality Assurance: Best Practices, Tools and Techniques for Software Developers. J. Ross Publishing, 2011.
- G. O'Regan: Introduction to Software Quality. Springer, 2014.
- W. Reif: Software-Verifikation und ihre Anwendungen, it+ti Themenheft, Oldenbourg Verlag, 1997
- Vorlesungsskript
- In der Vorlesung bereitgestellte wiss. Publikationen, Journalartikel und Buchbeiträge.

**Modulteil: Qualitätssicherung im Software Engineering (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Prüfung**

**Qualitätssicherung im Software Engineering (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

<b>Modul INF-0217: Praktikum Autonomes Fahren</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an den Modulveranstaltungen ist der Studierende in der Lage die Grundlagen des Autonomen Fahrens zu verstehen, anzuwenden und zu bewerten.</p> <p><b>Schlüsselqualifikationen:</b> Erlernen des eigenständigen Arbeitens mit Lehrbüchern (oder englischsprachiger Fachliteratur); Erwerb von Abstraktionsfähigkeiten</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 150 Std. Praktikum (Präsenzstudium)</p>		
<p><b>Voraussetzungen:</b> Empfohlen wird die Teilnahme an einem der beiden Seminare.  Modul Seminar Grundlagen des Software Engineering für Automotive Systems (BA) (INF-0027) - empfohlen Modul Seminar Grundlagen des Software Engineering für Automotive Systems (MA) (INF-0040) - empfohlen</p>		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 10	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Praktikum Autonomes Fahren</b>		
<b>Lehrformen:</b> Praktikum		
<b>Sprache:</b> Deutsch / Englisch		
<b>SWS:</b> 10		
<p><b>Inhalte:</b></p> <p>In diesem Praktikum lernen die Teilnehmer, wie verschiedene ausgewählte Teilaspekte des Autonomen Fahrens umgesetzt, simuliert und analysiert werden können.</p> <p>In einem Einführungskurs werden die benötigten theoretischen Grundlagen für die Bearbeitung der Praxisaufgaben gelegt.</p> <p>Darüber hinaus lernen die Teilnehmer u.a. das im Automotive-Umfeld häufig eingesetzte Entwicklungswerkzeug „ADTF“ von Elektrobit und die darin verwendete Programmiersprache C++ kennen.</p> <p>Nach dem Einführungskurs sollen in Gruppen von zwei bis drei Teilnehmern mithilfe der genannten Werkzeuge autonome Fahrfunktionen umgesetzt werden.</p> <p>Die entwickelten Ergebnisse werden abschließend an 1:8 Fahrzeugmodellen, deren Sensorik realen Fahrzeugen sehr nahe kommt, demonstriert und ausgewertet.</p>		
<b>Zugeordnete Lehrveranstaltungen:</b>		
<b>Praktikum über Autonomes Fahren (Praktikum)</b>		
<p>Im Praktikum "Autonomes Fahren" setzen sich die Teilnehmer/innen intensiv mit aktuellen Problemstellungen des autonomen Fahrens auseinander. Hierbei werden sowohl Software als auch Hardware (heterogene Sensorik, ECUs) in Betracht gezogen. Als Entwicklungsplattform dienen 1:8 Fahrzeugmodelle der AUDI AG, welche im Rahmen des Audi Autonomous Driving Cup zum Einsatz kommen. Alle weiteren Informationen zur Veranstaltung unter <a href="http://www.audi-autonomous-driving-cup.com">www.audi-autonomous-driving-cup.com</a></p>		

---

**Prüfung**

**Praktikum Autonomes Fahren**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

<b>Modul INF-0219: Seminar Architektur- und Technologiekonzepte (MA)</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Die immer weiter fortschreitende Digitalisierung beschränkt sich nicht mehr nur auf die Automatisierung von (Produktions-)Prozessen, sondern weitet sich auf die Produkte von etablierten Unternehmen aus. Es geht darum digitale Produkte möglichst schnell umzusetzen um innovative Ideen zu testen und Marktanteile sichern zu können. Mit diesem Wandel ergeben sich neue Anforderungen an die einzusetzenden Software-Architekturen und Technologien – ein Beispiel für eine solche Software-Architecture ist der Begriff "Microservice Architecture". In diesem Seminar sollen Kernaspekte und Prinzipien moderner, digitaler Software-Architekturen beleuchtet und an ausgewählten Beispielen "hands on" verprobt werden.		
<b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Architektur- und Technologiekonzepte (MA)</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch / Englisch <b>SWS:</b> 2		
<b>Inhalte:</b> In diesem Seminar sollen Kernaspekte und Prinzipien moderner, digitaler Software-Architekturen beleuchtet und an ausgewählten Beispielen "hands on" verprobt werden.		
<b>Literatur:</b> Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.		
<b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar		

<b>Modul INF-0232: Seminar Medical Information Sciences (MA)</b>		4 ECTS/LP
Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet der Medical Information Sciences selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.		
<b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken; Abwägen von Lösungsansätzen		
<b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Medical Information Sciences (Seminar)</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>SWS:</b> 2
<b>Inhalte:</b> Dieses Seminar soll die Grundlagen der Medical Information Sciences behandeln. Es sind verschiedene Themen zu bearbeiten die als Grundlage für ein nachfolgendes Praktikum dienen sollen.
<b>Literatur:</b> Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Seminar zu Medical Information Sciences f. Master (Seminar)</b> Bestandteil dieses Seminars sind fortgeschrittene Ansätze und Techniken im Bereich Medical Information Science.

<b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar
---

<b>Modul INF-0233: Industrierobotik</b>		8 ECTS/LP
Version 2.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage Industrieroboter zu programmieren. Sie können fachliche Lösungskonzepte in Roboterprogramme umsetzen und dabei Entwurfsalternativen bewerten und anwenden. Sie haben Kenntnisse von praxisrelevanten Aufgabenstellungen in der automatisierten Fertigung. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken.		
<b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis		
<b>Bemerkung:</b> Dieses Modul ersetzt das Modul "INF-0132: Software in Mechatronik und Robotik". Sofern dieses bereits belegt wurde, ist eine erneute Belegung <b>nicht</b> möglich.		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Industrierobotik (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> Ziel der Veranstaltung ist es, an Beispielen die Programmierung und der Entwurf von Software für Industrieroboter, wie sie z.B. in der Automobilindustrie verwendet werden, zu erlernen. Dazu werden im ersten Teil der Vorlesung Grundlagen wie Kinematik und Bahnplanung mit Hilfe simulationsbasierter Programmieransätze behandelt. Im zweiten Teil der Vorlesung werden verschiedene, kleine Programmieraufgabenstellungen bearbeitet und auf einem KUKA Roboter evaluiert. Die Programmierung erfolgt mit der Roboterprogrammiersprache KRL.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• L. Sciavicco, B. Siciliano: Modelling and Control of Robot Manipulators. Reihe: Advanced Textbooks in Control and Signal Processing. Springer 2000 (2nd Ed.)</li> <li>• Handbücher von KUKA</li> <li>• Folienhandout</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Industrierobotik (Vorlesung)</b> Diese Veranstaltung steht unter dem Motto "Uni goes Industry", da in dieser Veranstaltung die Programmierung "echter" Industrieroboter incl. zugehöriger Software-Entwicklungsumgebung und Simulationsmöglichkeiten		

vermittelt wird. Dazu werden in Zweiergruppen verschiedene, kleine Programmieraufgabenstellungen bearbeitet und auf einem KUKA KR 6 Roboter evaluiert. Zudem werden wichtige Grundlagen der Robotik wie Kinematik und Bahnplanung anhand eines simulierten Roboters behandelt. Ziel der Veranstaltung ist es, an Beispielen die Programmierung und der Entwurf von Software für Industrieroboter, wie sie z.B. in der Automobilindustrie verwendet werden, zu erlernen. Die Vorlesung "Industrierobotik" ersetzt die Veranstaltung "Software in Mechatronik und Robotik". Studierende, die an der Vorlesung "Software in Mechatronik und Robotik" teilgenommen haben, können diese Veranstaltung nicht mehr besuchen.  
... (weiter siehe Digicampus)

**Modulteil: Industrierobotik (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 4

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Industrierobotik (Übung)**

**Prüfung**

**Industrierobotik (mündliche Prüfung)**

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 4 Monate



<b>Modul INF-0235: Software für Industrie 4.0</b>		5 ECTS/LP
Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<b>Inhalte:</b> Die Vorlesung behandelt aktuelle Themen im Kontext von Industrie 4.0. Dazu zählen folgende Inhalte: <ul style="list-style-type: none"> <li>• Einführung in die Automatisierung</li> <li>• Referenzarchitekturen für Industrie 4.0</li> <li>• Einführung in die (mobile Service) Robotik</li> <li>• OPC UA</li> <li>• AutomationML</li> <li>• Data Analytics für Industrie 4.0</li> </ul>		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in Lage die Herausforderungen, die Industrie 4.0 für die Automatisierung und deren Softwareentwicklung bedeutet, zu verstehen. Sie werden sowohl in Techniken der klassischen Automatisierung als auch modernen Softwaretechnologien im Umfeld von Industrie 4.0 eingeführt. Sie sind in der Lage Lösungskonzepte zu erstellen und dabei Entwurfsalternativen bewerten und anwenden. Sie haben Kenntnisse von praxisrelevanten Aufgabenstellungen in der automatisierten Fertigung.  <b>Schlüsselqualifikationen:</b> Analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Übung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Moduleile</b>
<b>Moduleil: Software für Industrie 4.0 (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2
<b>Inhalte:</b> <ul style="list-style-type: none"> <li>• Einführung in die Automatisierung</li> <li>• Referenzarchitekturen für Industrie 4.0</li> <li>• Einführung in die (mobile Service) Robotik</li> <li>• OPC UA</li> <li>• AutomationML</li> <li>• Data Analytics für Industrie 4.0</li> </ul>

**Literatur:**

- Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0 - Abschlussbericht des Arbeitskreises Industrie 4.0; acaTech

**Modulteil: Software für Industrie 4.0 (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Prüfung**

**Prüfung Software für Industrie 4.0**

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

<b>Modul INF-0248: Kollaborative Robotik</b>	8 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Wolfgang Reif	
<p><b>Inhalte:</b></p> <p>Die Digitalisierung dringt mittlerweile in viele Bereiche des täglichen Lebens vor. Von Industriemaschinen und Robotersystemen (<i>Smart Factory, Industrie 4.0</i>) bis hin zu intelligenten Hausgeräten oder Heizungsanlagen (<i>Smart Home</i>) findet eine immer stärkere Vernetzung dieser Geräte im Internet der Dinge statt. Hierbei spricht man von cyber-physischen Systemen, die einerseits durch ein komplexes Zusammenspiel von vernetzten eingebetteten Systemen entstehen und andererseits geprägt sein werden von einer vollkommen neuen Art der Mensch-Technik-Interaktion in den Anwendungen.</p> <p>Eine besondere Rolle unter den cyber-physischen Systemen nehmen Assistenz- oder Serviceroboter ein, da sie durch einen geteilten Arbeitsraum in eine direkte, auch physische Interaktion mit dem Menschen treten können. Diese robotischen Systeme können den Menschen in seiner täglichen Arbeit unterstützen bzw. entlasten. Das gilt sowohl im häuslichen Umfeld als auch in der Fabrik. Im Kontext von Industrie 4.0 haben robotische Co-Worker (CoBots) vereinzelt bereits Eingang in die Produktion gefunden. Diese kollaborativen Einsatzszenarien werden jedoch in Zukunft noch weiter an Bedeutung gewinnen. Durch die in 2016 veröffentlichte Norm ISO/TS 15066:2016 zur Mensch-Roboter-Kollaboration stehen dem Einsatz innovativer Robotersysteme neue Möglichkeiten offen.</p> <p>Während die grundlegende Hardware, d.h. die Roboter, mobilen Plattformen, Greifer und Sensoren, bereitgestellt wird, liegt der Schwerpunkt der Veranstaltung auf der Entwicklung von Software für kollaborative (mobile) Manipulatoren. Durch die intelligente Vernetzung von Aktuatorik und Sensorik, neuartige Algorithmen für die Sensorauswertung und kreative Konzepte zur (physischen) Interaktion mit dem Menschen entstehen innovative, kollaborative Robotersysteme im praktischen Teil der Veranstaltung.</p> <p>Infolgedessen liegt der thematische Fokus der Veranstaltung auf</p> <ul style="list-style-type: none"> <li>• der Umsetzung multi-modaler Benutzerschnittstellen,</li> <li>• der Gestaltung neuartiger, auch physischer Interaktionsmöglichkeiten,</li> <li>• der Realisierung intelligenter Verhaltensweisen durch Techniken der Künstlichen Intelligenz bzw. des Machine Learnings und</li> <li>• der Entwicklung kognitiver Fähigkeiten des Robotersystems.</li> </ul> <p>Diese Kernelemente bieten die Grundlage, um eine praktische Aufgabenstellung prototypisch umzusetzen. Die Programmierung erfolgt dabei in objektorientierten Programmiersprachen. So können kollaborative Roboter bspw. in Java mit der Robotics API programmiert werden, einem am ISSE in Kooperation mit KUKA entwickelten Roboterframework. Alternativ oder ergänzend kann das Robot Operating System (ROS) verwendet werden, das in den Programmiersprachen C++ und Python programmiert werden kann.</p>	
<p><b>Lernziele/Kompetenzen:</b></p> <p>Die Studierenden sind in Lage kollaborative Roboter zu programmieren und kennen die regulatorischen Randbedingungen (vgl. ISO/TS 15066:2016). Sie kennen die grundlegenden Konzepte und Methoden der kollaborativen Robotik. Sie können dementsprechend fachliche Lösungskonzepte umsetzen und Entwurfsalternativen bewerten und anwenden. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken und können grundlegende Bewertungen der Risiken eines kollaborativen Robotersystems durchführen.</p> <p><b>Schlüsselqualifikationen:</b></p> <p>Analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis, Erwerb von Grundlagen für die Sicherheitsbewertung.</p>	
<p><b>Bemerkung:</b></p> <p><b>Voraussetzungen (empfohlen):</b></p> <p>Software für Industrie 4.0 oder Industrierobotik oder Grundlagen der Robotik</p>	
<p><b>Arbeitsaufwand:</b></p> <p>Gesamt: 240 Std.</p>	

<b>Voraussetzungen:</b> keine		<b>ECTS/LP-Bedingungen:</b> Bestehen der Modulprüfung
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Moduleile</b>
<p><b>Modulteil: Kollaborative Robotik (Vorlesung)</b>  <b>Lehrformen:</b> Vorlesung  <b>Sprache:</b> Deutsch  <b>SWS:</b> 2</p>
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• ISO/TS 15066:2016. 2016-02-15. Robots and robotic devices - Collaborative robots</li> <li>• DIN EN ISO 13849-1:2016-06. 06.2016. Sicherheit von Maschinen - Sicherheitsbezogene Teile von Steuerungen - Teil 1: Allgemeine Gestaltungsleitsätze (ISO 13849-1:2015); Deutsche Fassung EN ISO 13849-1:2015</li> <li>• DIN EN ISO 10218-1:2012. 01.2012. Industrieroboter - Sicherheitsanforderungen - Teil 1: Roboter (ISO 10218-1:2011); Deutsche Fassung EN ISO 10218-1:2011</li> </ul>
<p><b>Modulteil: Kollaborative Robotik (Übung)</b>  <b>Lehrformen:</b> Übung  <b>Sprache:</b> Deutsch  <b>SWS:</b> 4</p>

<p><b>Prüfung</b>  <b>Prüfung Kollaborative Robotik</b>  praktische Prüfung / Prüfungsdauer: 60 Minuten  Bearbeitungsfrist: 4 Monate</p>
--

<b>Modul INF-0077: Suchmaschinen</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Dr. Markus Endres		
<b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an der Veranstaltung sind die Studierenden in der Lage die Konzepte und Methoden, Verfahren, Techniken und Technologien von Suchmaschinen zu verstehen und zu bewerten. Darüber hinaus können die Studierenden weiterführende komplexe Problemstellungen auf dem Gebiet Datenbanken, insbesondere unter Verwendung von Präferenz-Suchmaschinen, analysieren und bewerten. Außerdem können die Studierenden fachliche Lösungskonzepte für Suchtechnologien in Programme umsetzen.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> Modul Datenbanksysteme (INF-0073) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig (i. d. R. im SoSe)	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Suchmaschinen (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 4		
<b>Inhalte:</b> Die Vorlesung behandelt grundlegende Konzepte von Suchmaschinen, Volltext-Suche, SQL-Suchmaschinen und Präferenz-Suchmaschinen (Preference SQL) sowie deren Implementierung.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• M. Levene: An Introduction to Search Engines and Web Navigation</li> <li>• R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval</li> <li>• I. H. Witten, M. Gori, T. Numerico: Web Dragons</li> <li>• W. Kießling: Foundations of Preferences in Database Systems</li> <li>• W. Kießling: Preference Queries with SV-Semantics</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Suchmaschinen (Vorlesung)</b> Die Vorlesung behandelt grundlegende Konzepte von Suchmaschinen sowie deren Implementierung.		
<b>Modulteil: Suchmaschinen (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		

---

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Suchmaschinen** (Übung)

**Prüfung**

**Suchmaschinen (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

<b>Modul INF-0078: Datenbankprogrammierung (Oracle)</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Dr. Markus Endres		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an der Veranstaltung sind die Studierenden in der Lage vertiefte Kenntnisse in Oracle anzuwenden. Darüber hinaus können die Studierenden komplexe, praxisrelevante Problemstellungen auf dem Gebiet Datenbanken, insbesondere unter Verwendung von Oracle, analysieren, bewerten und lösen. Sie kennen die Vor- und Nachteile unterschiedlicher ER-Modellierungen und können durch logisches und konzeptionelles Denken eine geeignete Lösung für komplexe Problemstellungen schaffen.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten,.</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
<b>Voraussetzungen:</b> Modul Datenbanksysteme (INF-0073) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Datenbankprogrammierung (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> Die Vorlesung behandelt Problemlösungsstrategien unter Zuhilfenahme einer Oracle-Datenbank. Dazu werden die Oracle-Architektur, Zugriffsrechte, Transformation von ER nach SQL, Oracle SQL, Aktive Inhalte wie PL/SQL und Java in Oracle, XML-Unterstützung in Oracle, Baumstrukturen, Tuning, Backup und Recovery behandelt.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• R. Elmasri, S. Navathe: Fundamentals of Database Systems</li> <li>• S. Melton: Understanding the New SQL: A Complete Guide</li> <li>• Oracle 11g Online-Dokumentation</li> </ul>		
<b>Modulteil: Datenbankprogrammierung (Übung)</b>		
<b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Prüfung</b>		
<b>Datenbankprogrammierung (Klausur)</b> Klausur / Prüfungsdauer: 60 Minuten		

<b>Modul INF-0080: Projektmodul Datenbanken und Informationssysteme</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Dr. Markus Endres		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet Datenbanken und Informationssysteme zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Selbständige Arbeit, Zeitmanagement, Eigenständige Literaturrecherche zu angrenzenden Themen, Arbeit mit englischsprachiger Fachliteratur, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
<p><b>Voraussetzungen:</b> Modul Datenbanksysteme (INF-0073) - empfohlen Modul Suchmaschinen (INF-0077) - empfohlen</p>		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Moduleile</b>
<p><b>Modulteil: Projektmodul Datenbanken und Informationssysteme</b>  <b>Lehrformen:</b> Praktikum  <b>Sprache:</b> Deutsch  <b>SWS:</b> 1</p>
<p><b>Inhalte:</b> Arbeiten am Präferenz-SQL-System des Lehrstuhls</p>
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Aktuelle Forschungsbeiträge zum Thema "Präferenzen"</li> <li>• Handbücher</li> </ul>

<p><b>Prüfung</b>  <b>Softwareabnahme, Vortrag, Abschlussbericht</b>          Praktikum</p>
---



<b>Modul INF-0092: Multimedia II: Machine Learning and Computer Vision</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<b>Lernziele/Kompetenzen:</b> Teilnehmer dieser Vorlesung beherrschen wichtige Konzepte des maschinellen Lernens, der Datenreduktion, der fortgeschrittenen Bildverarbeitung und des maschinellen Sehens und können diese anwenden. <b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Multimedia II: Machine Learning and Computer Vision (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 4
<b>Inhalte:</b> Die Vorlesung gibt einen guten Überblick über alle Aspekte des maschinellen Lernens und der maschinellen Extraktion von Informationen aus Multimediadaten (z.B. "Google Image Search", "Google Goggles"). Die erlernten Konzepte werden in den Übungen anhand von erfolgreichen Beispielen aus der Praxis ausprobiert, geübt, analysiert und bewertet. Zum Ende des Semesters werden fortgeschrittene Themen wie Objektdetektion und Objekterkennung von Gesichtern und Menschen behandelt. Die Inhalte der Vorlesung umfassen: Machine Learning (Decision Tree Learning, Artificial Neural Networks, Bayesian Learning, Discrete Adaboost), Data Reduction (Quantization (K-Means Clustering, Affinity Propagation), Dimensionality Reduction Techniques (PCA, NMF, Random Projection, MDS)) und Image Processing & Computer Vision (Salient Feature Points and Feature Descriptors, Object Detection (Face/Car/People Detection), Object Recognition (Face Recognition), Image Search with pLSA)
<b>Literatur:</b> Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Multimedia II (Vorlesung)</b> This course addresses state-of-the-art computer vision algorithms that let computers see, learn, and understand image and video content. After being taught the required basics in machine learning, students will - accompanied by practical exercises - get to know the most promising techniques. The topics of the course may be summarized as follows: Machine learning Implementation of algorithms using Intel IPP and MKL Image/video processing Media content analysis The learned concepts will be illustrated by successful examples in practice. The accompanying

exercises will contain some hands-on experiences. Towards the end of the course more advanced topics in object detection and object recognition will be addressed.

**Modulteil: Multimedia II: Machine Learning and Computer Vision (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Multimedia II (Übung)**

**Prüfung**

**Multimedia II: Machine Learning and Computer Vision (Klausur)**

Klausur / Prüfungsdauer: 120 Minuten

<b>Modul INF-0117: Funktionale Modellierung für Geoinformationssysteme</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller Prof. Dr. Sabine Timpf		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden verfügen über ein Verständnis der Grundlagen von Geoinformationssystemen. Sie wissen, wie deren Konzepte ohne Detailkenntnis von Programmiersprachen wie Java auf einfache, elegante und effektive Weise in einer funktionalen Programmiersprache abgebildet werden können. Sie haben diese Techniken anhand einer größeren Fallstudie validiert und können sie somit in konkreten Fragestellungen anwenden.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Funktionale Modellierung für Geoinformationssysteme (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Geometrien und Koordinaten, Projektionen und Transformationen, Vektor- und Rastermodelle, Topologien, Thematiken, Dynamik, räumliche Analyse, Map Algebra, Geo-Datenbanken, Coverage, spezielle Modellierungstechniken für Geodaten, Grundlagender funktionalen Programmierung und Modellierung, Fallstudie: Verkehrsnetz		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Eigenes Skriptum</li> <li>• B O'Sullivan, D. Stewart, J. Goerzen: Real World Haskell, O'Reilly 2008</li> <li>• M.Worboys, M. Duckham: GIS - A computing perspective, Routledge 2004</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Funktionale Modellierung für Geoinformationssysteme (Vorlesung)</b>		

**Modulteil: Funktionale Modellierung für Geoinformationssysteme (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Funktionale Modellierung für Geoinformationssysteme (Übung)**

Übung zu Funktionale Modellierung für Geoinformationssysteme

**Prüfung**

**Funktionale Modellierung für Geoinformationssysteme (Klausur)**

Klausur / Prüfungsdauer: 120 Minuten

<b>Modul INF-0118: Seminar Programmiermethodik und Multimediale Informationssysteme für Master</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "Programmiermethodik und Multimediale Informationssysteme" selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken, Literaturrecherche, Arbeit mit englischer Fachliteratur, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> in der Regel mind. 1x pro Studienjahr	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Moduleile</b>
<p><b>Moduleil:</b> Seminar Programmiermethodik und Multimediale Informationssysteme für Master</p> <p><b>Lehrformen:</b> Seminar</p> <p><b>Sprache:</b> Deutsch</p> <p><b>SWS:</b> 2</p>
<p><b>Inhalte:</b> Themen aus den Bereichen "Theoretische Informatik", "Multimedia" oder "Datenbanken und Informationssysteme"</p>
<p><b>Literatur:</b> wird jeweils bekanntgegeben</p>
<p><b>Zugeordnete Lehrveranstaltungen:</b> Seminar über Theoretische Informatik (Seminar)</p>

<p><b>Prüfung</b> Vortrag und schriftliche Ausarbeitung Seminar</p>
---

<b>Modul INF-0119: Projektmodul Programmiermethodik und Multimediale Informationssysteme</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet "Programmiermethodik und Multimediale Informationssysteme" zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; Bearbeitung konkreter Fallbeispiele; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis; Durchhaltevermögen; Erlernen von Präsentationstechniken; schriftliche Präsentation eigener Ergebnisse</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteil:</b> Projektmodul Programmiermethodik und Multimediale Informationssysteme
<b>Lehrformen:</b> Praktikum
<b>Sprache:</b> Deutsch
<b>SWS:</b> 1
<b>Inhalte:</b> Anwendung und Erweiterung von Kleene-Algebren, Halbringtheorie und automatisches Beweisen, Datenbanken und Informationssysteme
<b>Literatur:</b> aktuelle Forschungspaper

<b>Prüfung</b> Projektabnahme, Vortrag und Abschlussbericht Praktikum
---

<b>Modul INF-0213: Parallele und Verteilte Datenbanksysteme</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Dr. Markus Endres		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme an der Veranstaltung sind die Studierenden in der Lage die Konzepte und Methoden, Verfahren, Techniken und Technologien von parallelen und verteilten Datenbanksystemen zu verstehen und zu bewerten. Darüber hinaus können die Studierenden weiterführende komplexe Problemstellungen auf dem Gebiet Datenbanken, insbesondere unter Verwendung von parallelen und verteilten Algorithmen, analysieren und lösen.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten.</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 3.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<b>Modulteil: Parallele und Verteilte Datenbanksysteme (Vorlesung)</b>		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<p><b>Inhalte:</b> Die Vorlesung behandelt parallele und verteilte Datenbanksysteme. Dazu wird die Architektur von parallelen und verteilten Datenbanksystemen vorgestellt. Der Schwerpunkt liegt anschließend bei der Ausarbeitung und Implementierung von parallelen und verteilten Algorithmen. Insbesondere werden hierbei die parallele Suche, parallele Gruppierung, parallele Joins, Transaktionen in verteilten Datenbanken, verteilte Transaktionsprotokolle, paralleles OLAP, paralleles Data Mining und paralleles Clustering und Klassifikation behandelt.</p>		
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• - Taniar et al.: High-Performance Parallel Database Processing and Grid Databases</li> <li>• - E. Rahm: Mehrrechner-Datenbanksysteme</li> <li>• - M. T. Özsu, P. Valduriez: Principles of Distributed Database Systems</li> <li>• - P. Dada: Verteilte Datenbanken und Client/Server-Systeme</li> <li>• - S. Conrad: Förderierte Datenbanksysteme</li> <li>• - S. K. Rahimi, F. S. Haug: Distributed Database Management Systems - A Practical Approach</li> </ul>		
<b>Modulteil: Parallele und Verteilte Datenbanksysteme (Übung)</b>		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		

---

**Prüfung**

**Parallele und Verteilte Datenbanksysteme (mdl. Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten



<b>Modul INF-0227: Seminar Datenbanksysteme für Master</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Dr. Markus Endres		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien aus dem Gebiet Datenbanken zu verstehen und bezogen auf ein spezielles Thema aus dem genannten Gebiet zu bewerten.</p> <p>Sie verfügen über die Arbeitstechniken, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Kommunikationsfähigkeit; Fähigkeit zum Einsatz neuer Medien; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Präsentationstechniken</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<b>Voraussetzungen:</b> Modul Datenbanksysteme (INF-0073) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig (i. d. R. im SoSe)	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Datenbanksysteme für Master</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>Angebotshäufigkeit:</b> jedes Sommersemester		
<b>SWS:</b> 2		
<b>Inhalte:</b> Aktuelle Forschungsbeiträge aus den Bereich "Datenbanken und Informationssysteme".		
<b>Literatur:</b> Aktuelle Forschungsbeiträge		
<b>Zugeordnete Lehrveranstaltungen:</b>		
<p><b>Seminar Datenbanksysteme</b> (Seminar) Aktuelle Forschungsbeiträge aus dem Bereich "Datenbanken und Informationssysteme". Das Seminar behandelt Konzepte und Techniken der Implementierung von Datenbanksystemen. Insbesondere werden Themen wie Datenstrukturen, Algorithmen, Speicher- und Indexstrukturen, Anfrageverarbeitung und -optimierung, Transaktionsverwaltung und Recovery behandelt. In der Einführungsveranstaltung am 25.04.2017 um 15:45 Uhr in Raum 2056 N klären wir organisatorische Details und verteilen die Vortragsthemen unter den Teilnehmern. Das Seminar gehört zum Bereich Datenbanken und Informationssysteme. Die Veranstaltung richtet sich sowohl an Studierende im Bachelor- als auch im Masterstudium. Das Seminar wird als Blockseminar stattfinden. Die genauen Termine werden noch bekannt gegeben.</p>		

---

**Prüfung**

**Vortrag und schriftliche Ausarbeitung**

Seminar

<b>Modul INF-0240: Seminar Informationssysteme für Master</b>		4 ECTS/LP
Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Dr. Markus Endres		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien aus dem Gebiet Informationssysteme zu verstehen und bezogen auf ein spezielles Thema aus dem genannten Gebiet zu bewerten.</p> <p>Sie verfügen über die Arbeitstechniken, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Kommunikationsfähigkeit; Fähigkeit zum Einsatz neuer Medien; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Präsentationstechniken</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> Modul Datenbanksysteme (INF-0073) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig (i. d. R. im WS)	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Informationssysteme für Master</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Aktuelle Forschungsbeiträge aus den Bereich "Datenbanken und Informationssysteme".		
<b>Literatur:</b> Aktuelle Forschungsbeiträge		
<b>Prüfung</b>		
<b>Vortrag und schriftliche Ausarbeitung</b> Seminar		

<b>Modul INF-0067: Peer-to-Peer und Cloud Computing</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<b>Lernziele/Kompetenzen:</b> Erwerb fundierter Kenntnisse über Konzepte und Anwendungen von Cloud-Computing bzw. Peer-to-Peer-Systemen als Grundlage komplexer Internet basierter Infrastrukturen. Dazu werden ein Verständnis für Probleme beim Entwurf von komplexen vernetzten Systemen erarbeitet und forschungsorientierte Lösungsansätze vermittelt.		
<b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Peer-to-Peer und Cloud Computing (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Die Vorlesung "Cloud- und Peer-to-Peer-Computing" vermittelt Ansätze zur Organisation von komplexen abstrahierten IT-Infrastrukturen, die dynamisch an wechselnde Nutzungsbedingungen angepasst werden können und Dienste auf verschiedenen Ebenen zur Verfügung stellen, z.B. Rechenkapazität, Datenspeicher, Netzkapazitäten und Softwaredienste. Dazu werden zunächst Anforderungen und Ziele solcher Systeme definiert und diskutiert. Darüber hinaus werden Konzepte aus dem Bereich der Systemarchitekturen und Ansätze aus dem Bereich selbstorganisierender Algorithmen dargestellt und bewertet. In allen Teilen werden Bezüge zu konkreten Anwendungsgebieten gegeben. Die zugehörige Übung bietet die Möglichkeit, die erlernten Ansätze zu vertiefen und beispielhaft anzuwenden.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• aktuelle wissenschaftliche Paper</li> <li>• Mahlmann und Schindelhauer: Peer-to-Peer Netzwerke - Algorithmen und Methoden, Springer 2007</li> <li>• Antonopoulos und Gillam: Cloud Computing - Principles, Systems and Applications, Springer 2010</li> </ul>		
<b>Modulteil: Peer-to-Peer und Cloud Computing (Übung)</b>		
<b>Lehrformen:</b> Übung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		

---

**Prüfung**

**Peer-to-Peer und Cloud Computing (mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

**Beschreibung:**

Die Prüfung kann jedes Semester zu Beginn und Ende der vorlesungsfreien Zeit abgelegt werden.

<b>Modul INF-0069: Weiterführende Betriebssystemkonzepte</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<b>Lernziele/Kompetenzen:</b> Erwerb weiterführender Kenntnisse über das Forschungsgebiet Betriebssysteme, basierend auf grundlegenden Konzepten der systemnahen Informatik und Betriebssystemen. Dazu wird ein Verständnis für Probleme bei der Entwicklung moderner Betriebssysteme erarbeitet und anhand von Beispielen illustriert. Die erworbenen Kenntnisse werden anhand von praktischen Übungen vertieft.		
<b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Weiterführende Betriebssystemkonzepte (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Die Vorlesung "Weiterführende Betriebssystemkonzepte" vermittelt aufbauend auf den grundlegenden Mechanismen, die bereits aus der Vorlesung "Systemnahe Informatik" bekannt sind, Einblicke in die Funktionsweise von modernen Betriebssystemen. Dabei wird der Fokus des theoretischen Teils auf dem Verständnis von Basismechanismen unter anderem aus den Bereichen Scheduling, Memorymanagement und Input/Output stehen. Der praktische Teil konzentriert sich dabei auf die Umsetzung unterschiedlicher Techniken im Labormaßstab sowie die Evaluation der Leistungsfähigkeit dieser implementierten Konzepte. Grundlage der Arbeiten sind dabei aktuelle Betriebssysteme beispielsweise aus dem Umfeld der Linux und Android Systeme.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Folien</li> <li>• Andrew S. Tanenbaum: "Moderne Betriebssysteme", Pearson Studium, ISBN:978-3-8273-7342-7</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Weiterführende Betriebssystemkonzepte (Vorlesung)</b> (Vorlesung) Die Vorlesung „Weiterführende Betriebssystemkonzepte“ vermittelt weiterführende Ansätze zur Funktionsweise von Betriebssystemen in verschiedenen Anwendungsgebieten. Dazu werden zunächst Anforderungen und Ziele solcher Systeme definiert und diskutiert. In dieser Vorlesung werden vor allem Besonderheiten der Betriebssysteme aus dem Bereich Sensornetze (z.B. TinyOS), mobile Endgeräte (z.B. Android) und klassischer Desktop- und Serverumgebungen (z.B. Linux) betrachtet und verglichen. Dazu werden beispielsweise Aspekte		

aus den Bereichen Speicherverwaltung, Dateisysteme, IO und Sicherheit betrachtet. In allen Teilen werden Bezüge zu konkreten Anwendungsgebieten gegeben. Die zugehörige Übung bietet die Möglichkeit, die erlernten Ansätze zu vertiefen und beispielhaft anzuwenden.

**Modulteil: Weiterführende Betriebssystemkonzepte (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 4

**Zugeordnete Lehrveranstaltungen:**

**Weiterführende Betriebssystemkonzepte (Übung)** (Übung)

**Prüfung**

**Weiterführende Betriebssystemkonzepte (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

<b>Modul INF-0071: Seminar Naturalloge Algorithmen und Multiagentensysteme</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, spezifische Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien im Schnittbereich naturalogener Verfahren und Multiagentensysteme selbstständig zu erarbeiten und bezogen auf ein spezielles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die Arbeitstechniken, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> selbstständiges wissenschaftliches Arbeiten</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Naturalloge Algorithmen und Multiagentensysteme</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<p><b>Inhalte:</b> In dem Seminar sollen sich die Studenten jeweils einem speziellen Thema in Schnittbereich naturaloge Algorithmen und Multiagentensysteme genauer beschäftigen. Diese Thema kann ein bestimmte Anwendung, z.B. das Swarmoid-Projekt, sein oder auch eine bestimmte Technik, z.B. für Task Allocation betreffen. Sie erstellen einen etwa 30-minütigen Vortrag zum gegebenen individuellen Thema. In einer schriftlichen Ausarbeitung werden die Erkenntnisse zum Thema zusammengefasst.</p>		
<p><b>Literatur:</b> wird im Seminar bekanntgegeben</p>		
<p><b>Zugeordnete Lehrveranstaltungen:</b> <b>Naturalloge Algorithmen und Multi-Agenten Systeme (Seminar)</b> (Seminar) Es handelt sich um eine Master-Veranstaltung. Es werden max. 10 Themen vergeben. Die Zuordnung eines Themas erfolgt per Lösung, aber unter Berücksichtigung von Präferenzen.</p>		
<p><b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar</p>		



<b>Modul INF-0084: Seminar Next Generation Networks</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe15) Modulverantwortliche/r: Prof. Dr. Rudi Knorr		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, ein wissenschaftlich anspruchsvolles Thema aus dem Gebiet "Next Generation Networks" selbständig zu erarbeiten, geeignet in Schrift und Vortrag zu präsentieren und sachlich über Vorträge zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Fähigkeit zur Beurteilung von Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten. Selbständige und wissenschaftliche Arbeitsweise.</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> Modul Kommunikationssysteme (INF-0081) - empfohlen		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Seminar Next Generation Networks</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>		
<p><b>Inhalte:</b> Die Themen für dieses Seminar werden jedes Jahr unter Berücksichtigung aktueller Trends aus dem Gebiet "Next Generation Networks" neu festgelegt.</p>		
<p><b>Literatur:</b> Grundlegende und aktuelle Forschungsliteratur in Abhängigkeit von den festgelegten Themen.</p>		
<p><b>Zugeordnete Lehrveranstaltungen:</b> <b>Seminar Next Generation Networks</b> (Seminar)</p>		
<p><b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar</p>		

<b>Modul INF-0085: Projektmodul Kommunikationssysteme</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rudi Knorr		
<b>Lernziele/Kompetenzen:</b> Die Studierenden haben tiefere Fachkenntnisse und Fähigkeiten zu "Kommunikationssysteme" erworben, die es ihnen ermöglichen, an die internationale Forschung anzuknüpfen. Sie sind fähig, innovative Methoden bei der Lösung von Problemen in diesem Gebiet anzuwenden und einen wissenschaftlichen Beitrag zu diesem Gebiet zu leisten.		
<b>Schlüsselqualifikationen:</b> selbständige und strukturierte Arbeitsweise, analytisch-methodische Kompetenz, Grundsätze guter wissenschaftlicher Praxis		
<b>Arbeitsaufwand:</b> Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Projektmodul Kommunikationssysteme</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 1		
<b>Inhalte:</b> Aktuelle Forschungsthemen auf dem Gebiet "Kommunikationssysteme".		
<b>Literatur:</b> wissenschaftliche Papiere, Handbücher		
<b>Prüfung</b> <b>Vortrag und Abschlussbericht</b> Praktikum		

<b>Modul INF-0145: Mikrorechnertechnik und Echtzeitsysteme</b>		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<b>Lernziele/Kompetenzen:</b> Die Studierenden besitzen fundierte Kenntnisse der Prinzipien des Aufbaus von Mikrocontrollern und deren Peripherie, der Konzepte gängiger Mikrocontroller, der Leistungsfähigkeit und Grenzen von Mikrocontrollern beim Einsatz in eingebetteten Systemen. Weiterhin kennen die Studierenden die Probleme und Lösungen, die für den Aufbau und die Funktionsweise von sicherheitskritischen Echtzeitsystemen nötig sind.		
<b>Schlüsselqualifikationen:</b> Analytisch-methodische Kompetenz in den Bereichen der Mikrorechnertechnik und Echtzeitsysteme, Abwägung von Lösungsansätzen, Präsentation von Lösungen von Übungsaufgaben.		
<b>Arbeitsaufwand:</b> Gesamt: 180 Std. 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 75 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 45 Std. Vorlesung (Präsenzstudium) 15 Std. Übung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Mikrorechnertechnik und Echtzeitsysteme (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 3		
<b>Inhalte:</b> Die Vorlesung "Mikrorechnertechnik und Echtzeitsysteme" behandelt die grundlegenden Prinzipien der Mikrocontroller. Es werden vertiefte Kenntnisse der Mikrocontroller und der Mikrocontroller-Komponenten bereitgestellt. In der Praxis häufig verwendete Mikrocontroller werden in ihrer Funktionsweise analysiert und zukunftsweisende Technologien dieser Bausteine erläutert. Ein weiterer Schwerpunkt der Vorlesung sind Echtzeitsysteme. Es werden die Herausforderungen von Echtzeitbedingungen auf die Prozessorarchitektur sowie Möglichkeiten ihnen zu begegnen betrachtet. Techniken der Echtzeitprogrammierung, Echtzeit-Scheduling, Echtzeitbetriebssysteme und der WCET-Analyse werden vermittelt. Schließlich werden die für eingebettete Echtzeit- und Automatisierungsanwendungen wichtigen Feldbusse (Profibus und CAN-Bus) besprochen. Zum Schluss wird in Automotive- und Avionics-Systeme eingeführt.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010</li> <li>• Heinz Wörn, Uwe Brinkschulte, Echtzeitsysteme, Springer Verlag, Berlin/Heidelberg, 2005</li> </ul>		
<b>Modulteil: Mikrorechnertechnik und Echtzeitsysteme (Übung)</b>		
<b>Lehrformen:</b> Übung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 1		

**Prüfung**

**Mikrorechnertechnik und Echtzeitsysteme (Klausur)**

Klausur / Prüfungsdauer: 60 Minuten

<b>Modul INF-0146: Cyber-Physical Systems</b>		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden besitzen fundierte Kenntnisse in der Modellierung, dem Entwurf und der Analyse eingebetteter Echtzeitsysteme. Sie kennen die Schlüsselprobleme, die in solchen Systemen auftreten können und sind mit entsprechenden Lösungsansätzen vertraut.</p> <p><b>Schlüsselqualifikationen:</b> Analytisch-methodische Kompetenz im Bereich der Cyber-Physical Systems, Abwägung von Lösungsansätzen, Präsentation von Lösungen von Übungsaufgaben</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 180 Std. 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 75 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 45 Std. Vorlesung (Präsenzstudium) 15 Std. Übung (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> Modul Systemnahe Informatik (INF-0138) - empfohlen		
<b>Angebotshäufigkeit:</b> wird nicht mehr angeboten!	<b>Empfohlenes Fachsemester:</b> 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Cyber-Physical Systems (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 3		
<p><b>Inhalte:</b> Die Vorlesung Cyber-Physical Systems befasst sich mit der Integration eingebetteter Systeme mit der physikalischen Welt. Dies erfolgt in drei Teilen: Der erste Teil befasst sich mit der Modellierung von physikalischen Vorgängen und Steuerungssystemem. Der zweite Teil behandelt den Entwurf eines Computers und insbesondere der notwendigen Software für ein System, das in physikalische Prozesse eingebettet ist und mit diesen in Rückkopplung steht. In diesem Teil werden wichtige Techniken für Echtzeitsysteme vorgestellt. Der dritte Teil der Vorlesung geht auf Analyse und Verifikation solcher Systeme ein. Hier werden Techniken besprochen, die insbesondere beim Entwurf sicherheitskritischer Systeme von Relevanz sind, etwa im Umfeld des Fahrzeugbaus oder der Luftfahrt.</p>		
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• E. A. Lee, S. A. Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, LeeSeshia.org, 2011</li> <li>• Jane W. S. Liu, Real-Time Systems, Prentice Hall, 2000</li> <li>• G.C. Buttazzo, Hard Real-Time Computing Systems, Second Edition, Springer, 2005</li> <li>• E. A. Lee, P. Varaiya, Structure and Interpretation of Signals and Systems, Second Edition, LeeVaraiya.org, 2011</li> </ul>		

---

**Modulteil: Cyber-Physical Systems (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 1

**Prüfung**

**Cyber-Physical Systems (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

<b>Modul INF-0147: Prozessorarchitektur</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden besitzen fundierte Kenntnisse über Prinzipien des Aufbaus von superskalaren Mikroprozessoren und Multicore-Prozessoren. Sie kennen und verstehen aktuelle Konzepte der Prozessorarchitektur und könne die Vor- und Nachteile aktueller und zukünftiger Prozessoren anhand ihres internen Aufbaus einschätzen.</p> <p><b>Schlüsselqualifikationen:</b> Analytisch-methodische Kompetenz im Bereich der Prozessorarchitektur, Abwägung von Lösungsansätzen, Präsentation von Lösungen von Übungsaufgaben.</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p>		
<b>Voraussetzungen:</b> Modul Systemnahe Informatik (INF-0138) - empfohlen		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Prozessorarchitektur (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> Die Vorlesung "Prozessorarchitektur" vertieft die Techniken superskalarer Mikroprozessoren und aktueller Multicore-Prozessoren. Dabei werden die Pipelineinstufen detailliert behandelt, mehrfädige Prozessoren und Multicores gegenübergestellt sowie aktuelle Beispielprozessoren vorgestellt. Außerdem wird aus der Forschung an Manycores und Echtzeit-Multicores berichtet.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010</li> <li>• John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b> Prozessorarchitektur (Vorlesung)		
<b>Modulteil: Prozessorarchitektur (Übung)</b>		
<b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Zugeordnete Lehrveranstaltungen:</b> Übung zu Prozessorarchitektur (Übung)		

Bitte melden Sie sich nur bei der zugehörigen Vorlesung, nicht bei dieser Übung an

**Prüfung**

**Prozessorarchitektur (Klausur)**

Klausur / Prüfungsdauer: 60 Minuten



<b>Modul INF-0148: Entwurf und Analyse fehlertolerierender Rechensysteme</b>		5 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<b>Lernziele/Kompetenzen:</b> Nach Abschluss der Vorlesung kennen und verstehen die Studierenden grundlegende Methoden und Verfahren im Bereich fehlertolerierender Rechensysteme. Sie wissen wo, wann und weshalb welche Redundanzarten zum Einsatz kommen und können die erlernten Konzepte in kleinerem Rahmen implementieren. Sie kennen verschiedene Methoden zur Bewertung und Modellierung von fehlertolerierenden Rechensystemen wie Wahrscheinlichkeitsrechnung, Fehlerbäume, Zuverlässigkeits-Blockdiagramme sowie Markovketten und können diese anwenden.		
<b>Schlüsselqualifikationen:</b> Abstraktionsvermögen, analytisch-methodisches sowie vernetztes Denken		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Übung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Entwurf und Analyse fehlertolerierender Rechensysteme (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> Der Vorlesung führt in den Entwurf und die Analyse fehlertolerierender Rechensysteme ein. Zunächst werden verschiedene Fehlerarten charakterisiert und die Bedeutung von Fehlermodellen hervorgehoben. Danach werden unterschiedliche Hardware- und Software-Methoden zur Erkennung und Tolerierung von Fehlern vorgestellt. Die diskutierten Maßnahmen beziehen sich nicht nur auf strukturelle, sondern auch auf zeitliche und informationelle Redundanz (fehlertolerierende Codes). Nach einem kurzen Repetitorium der Wahrscheinlichkeitsrechnung und Kombinatorik werden verschiedene Analysemethoden wie klassische Wahrscheinlichkeitsrechnung, Zuverlässigkeits-Blockdiagramme, Markovketten, FMEA und Fehlerbäume vorgestellt, Unterschiede hervorgehoben und anhand praktischer Beispiele erläutert. Um ein fehlertolerierendes System bewerten zu können, müssen Fehlerinjektionsexperimente durchgeführt werden. Aus diesem Grund werden verschiedene Möglichkeiten der Fehlerinjektion kurz angeschnitten.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• D.P. Siewiorek, R.S. Swarz: Reliable Computer Systems, Peters, 1998</li> <li>• I. Koren, C.M. Krishna: Fault Tolerant Systems, Morgan Kaufmann, 2007</li> <li>• T. Anderson, P.A. Lee: Fault Tolerance - Principles and Practice, Prentice Hall, 1982</li> </ul>		

---

**Modulteil: Entwurf und Analyse fehlertolerierender Rechensysteme (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Prüfung**

**Entwurf und Analyse fehlertolerierender Rechensysteme (Klausur)**

Klausur / Prüfungsdauer: 60 Minuten

<b>Modul INF-0149: Praktikum Eingebettete Systeme</b>		5 ECTS/LP
Version 1.2.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage Projektaufgaben zu einer Themenstellung aus dem Gebiet "Eingebettete Systeme" einzeln oder Team zu planen, nach einem selbst entwickelten fundierten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.		
<b>Schlüsselqualifikationen:</b> Projektgebundene Arbeit und Zeitmanagement.		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
<b>Voraussetzungen:</b> Kenntnisse in C. Modul Mikrorechner-technik und Echtzeitsysteme (INF-0145) - empfohlen		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Praktikum Eingebettete Systeme</b>		
<b>Lehrformen:</b> Praktikum		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 4		
<b>Inhalte:</b> In dem Praktikum "Eingebettete Systeme" sollen die Einschränkungen und Herausforderungen für das Programmieren von eingebetteten Systemen erlernt werden. Als Plattform dient ein Staubsaugerroboter (ROOMBA) und ein daran angeschlossener Mikrocontroller zur Steuerung des ROOMBA. Die Programmierung erfolgt hardwarenah in C und die erstellten Programme sollen die Sensoren des ROOMBA auslesen und entsprechende Aktuatoren stellen. Dabei sollen insbesondere die Herausforderungen eingebetteter Systeme, wie z.B. das Zeitverhalten der Software sowie das Arbeiten mit Datenblättern kennengelernt werden. In einer Projektphase sollen die anfänglich erlernten Grundkenntnisse vertieft werden und komplexere Steuerungsprogramme entwickelt werden, z.B. ein autonomer Explorator oder ein "ROOMBA-Rennen" durch ein Labyrinth. Die Projekte werden einzeln oder im Team bearbeitet, dokumentiert und am Ende des Praktikums präsentiert.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Marwedel, Wehmeyer: Eingebettete Systeme, Springer Verlag, Heidelberg, 2007</li> <li>• Wiegmann: Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller, 6. Auflage, VDE Verlag, Berlin, 2011</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b>		
<b>Eingebettete Systeme</b> (Praktikum)		
<b>Prüfung</b>		
<b>Erfolgreiche Praktikumsteilnahme, Projektvorstellung</b> Praktikum		

<b>Modul INF-0150: Hardware-Entwurf</b>		8 ECTS/LP
Version 1.0.0 (seit WS14/15) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage Projektaufgaben zu einer Themenstellung aus dem Gebiet Prozessorarchitektur im Team zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.		
<b>Schlüsselqualifikation:</b> Projektgebundene Arbeit und Zeitmanagement		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> Modul Systemnahe Informatik (INF-0138) - empfohlen		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Hardware-Entwurf (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2
<b>Inhalte:</b> Die Veranstaltung "Hardware-Entwurf" stellt Methoden des logischen Entwurfs digitaler Schaltungen dar, angefangen von der abstrakten Beschreibung in einer Hardwarebeschreibungssprache (wie VHDL) bis hin zur physikalischen Implementierung auf Transistorebene. Im praktischen Teil der Veranstaltung wird die Hardwareentwicklung am Beispiel einer fünfstufigen Prozessorpipeline veranschaulicht. Das Ergebnis ist ein in VHDL entwickelter lauffähiger Prozessor für ein FPGA-Prototypenboard
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010</li> <li>• John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011</li> </ul>
<b>Modulteil: Hardware-Entwurf (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 4

---

**Prüfung**

**Projektvorstellung und Projektabnahme**

Praktikum

**Beschreibung:**

Erfolgreiche Praktikumsteilnahme, Projektvorstellung am Ende des Semesters

<b>Modul INF-0151: Praktikum Multicore-Programmierung</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage Projektaufgaben zu einer Themenstellung aus dem Gebiet der parallelen Programmierung von Multicores im Team zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.		
<b>Schlüsselqualifikationen:</b> Projektgebundene Arbeit und Zeitmanagement		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Praktikum (Präsenzstudium)		
<b>Voraussetzungen:</b> Modul Multicore-Programmierung (INF-0139) - empfohlen		
<b>Angebotshäufigkeit:</b> wurde ersetzt durch INF-0216	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Praktikum Multicore-Programmierung</b>		
<b>Lehrformen:</b> Praktikum		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 4		
<b>Inhalte:</b> Techniken der Parallelprogrammierung und verschiedene APIs zur Parallelprogrammierung (MPI, GPU-Programmierung mit OpenCL, Boost Threads, transaktionaler Speicher)		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Thomas Rauber, Gundula Rüger: Parallele Programmierung, Springer Verlag 2007.</li> <li>• es werden die jeweils neuesten Java-, OpenCL- und Multicore-Unterlagen aus dem Internet verwendet</li> </ul>		
<b>Prüfung</b>		
<b>Projektvorstellung und Projektabnahme</b> Praktikum		

<b>Modul INF-0152: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet der Prozessorarchitekturen selbstständig zu erarbeiten, zu analysieren und bezogen auf das individuelle Seminarthema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz entsprechender Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren und zu bewerten.</p> <p><b>Schlüsselqualifikationen:</b> Grundsätze guter wissenschaftlicher Praxis, Zeitmanagement, Literaturrecherche, Arbeit mit englischsprachiger Fachliteratur</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Moduleile</b>
<p><b>Modulteil: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen</b></p> <p><b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>
<p><b>Inhalte:</b> Im Seminar werden Architekturen und Technologien moderner Prozessoren aus Forschung und Industrie behandelt. Jeder Seminarteilnehmer erhält individuelle Literaturhinweise, die dann im Laufe des Seminars durch weitere eigenständig erarbeitete Referenzen ergänzt werden sollen. Abschluss des Seminars stellt eine schriftliche Ausarbeitung sowie ein Vortrag über das behandelte Thema dar.</p>
<p><b>Literatur:</b> individuell gegeben und Selbstrecherche</p>
<p><b>Zugeordnete Lehrveranstaltungen:</b> <b>Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen (Seminar)</b> Der Schwerpunkt liegt dieses Semester auf neuartigen Technologien und wie sie die Prozessorarchitektur beeinflussen könnten. Mögliche Themen sind Nanotubes, Graphen, 3D-Stacking, Spintronik, Memristoren, Quantencomputer, künstliche neuronale Netze, nichtflüchtige Speicher, ...</p>

<p><b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar</p>
--

<b>Modul INF-0153: Seminar Safety-Critical Systems</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet sicherheitskritischer Systeme selbstständig zu erarbeiten, zu analysieren und bezogen auf das individuelle Seminarthema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren und zu bewerten.</p> <p><b>Schlüsselqualifikationen:</b> Grundsätze guter wissenschaftlicher Praxis, Zeitmanagement, Literaturrecherche, Arbeit mit englischsprachiger Fachliteratur</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<b>Modulteil: Seminar Safety-Critical Systems</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<p><b>Inhalte:</b> Im Seminar werden Themen aus dem Bereich der sicherheitskritischen Systeme behandelt. Jeder Seminarteilnehmer erhält individuelle Literaturhinweise, die dann im Laufe des Seminars durch weitere eigenständig erarbeitete Referenzen ergänzt werden sollen. Abschluss des Seminars stellt eine schriftliche Ausarbeitung sowie ein Vortrag über das behandelte Thema dar.</p>		
<p><b>Literatur:</b> individuell gegeben und Selbstrecherche</p>		
<b>Prüfung</b>		
<b>Vortrag und schriftliche Ausarbeitung</b>		
Seminar		



<b>Modul INF-0154: Projektmodul Systemnahe Informatik und Kommunikationssysteme</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage Problemstellungen höherer Komplexität auf dem Gebiet der Systemnahen Informatik zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Selbständige Arbeit, Zeitmanagement, Eigenständige Literaturrecherche zu angrenzenden Themen, Arbeit mit englischsprachiger Fachliteratur, Grundsätze guter wissenschaftlicher Praxis.</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Projektmodul Systemnahe Informatik und Kommunikationssysteme</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 1</p>		
<b>Inhalte:</b> Autonome Mitarbeit an aktuelle Forschungsthemen.		
<b>Literatur:</b> wissenschaftliche Papiere, Handbücher		
<b>Prüfung</b>		
<b>Vortrag und schriftliche Ausarbeitung</b> Praktikum		

<b>Modul INF-0216: Vertiefte Multicore-Programmierung</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage, vertiefte Themenstellungen aus dem Gebiet der parallelen Programmierung von Multicores im Team zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren. Sie lernen verschiedene Paradigmen der parallelen Programmierung einzuschätzen. <b>Schlüsselqualifikationen:</b> Abstraktes Denken, Projektgebundene Arbeit in Teams, Präsentation und Diskussion von Ergebnissen, Zeitmanagement		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Vertiefte Multicore-Programmierung (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Die Vorlesung vertieft verschiedene Techniken der Parallelprogrammierung für aktuelle Multicore-Prozessoren und Grafikkarten. Nach einer grundlegenden Einführung in Threads, Synchronisationskonstrukte und weiterführende Konzepte der Parallelprogrammierung in C++11 werden weitere parallele Programmiermodelle behandelt.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• M. Herlihy, N. Shavit: The Art of Multiprocessor-Programming, Morgan Kaufmann, 2012, ISBN: 978-0123973375</li> <li>• M. McCool, J. Reinders, A. D. Robison: Structured Parallel Programming: Patterns for Efficient Computation, Morgan Kaufmann, 2012, ISBN: 978-0124159938</li> <li>• T. Rauber, G. Rünger: Parallele Programmierung, Springer Verlag, 2007, ISBN: 978-3540465492</li> <li>• es werden die jeweils neuesten APIs/Unterlagen aus dem Internet verwendet</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b>		
<b>Vertiefte Multicore-Programmierung (Vorlesung)</b> In der Vorlesung und Übung werden die folgenden Technologien behandelt: - C++11 Threads - OpenMP - Transaktionaler Speicher (Intel TSX) - Nachrichtengekoppelte Programmierung (Message Passing Interface, MPI) - GPU-Programmierung (CUDA)		

**Modulteil: Vertiefte Multicore-Programmierung (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 4

**Inhalte:**

Ziel der Übung ist es, den Umgang mit den unterschiedlichen Programmiermodellen sowie Performanzanalyse- und Debugging-Techniken in praktischen Beispielen zu vertiefen. Die Übung wird durch eine Projektphase abgeschlossen, die es den Studierenden ermöglicht, die behandelten Programmier Techniken in einem umfangreicheren Projekt selbständig anzuwenden, Ergebnisse auszuwerten und zu präsentieren.

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Vertiefte Multicore-Programmierung (Übung)**

**Prüfung**

**Vertiefte Multicore-Programmierung (mündliche Prüfung)**

Mündliche Prüfung, Dauer: 30-45 Minuten

**Beschreibung:**

Projektvorstellung und -abnahme, Fragen zu Vorlesung und Übung

<b>Modul INF-0050: Constrained data structures</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<b>Lernziele/Kompetenzen:</b> Verständnis für Ziele des Entwurfs von Datenstrukturen; die Fertigkeit, Datenstrukturen für unterschiedliche Anwendungen zu konstruieren und zu analysieren sowie sinnvoll zwischen verschiedenen Datenstrukturen für gegebene Aufgaben zu wählen; Erfahrung im Bereich der Algorithmen und Datenstrukturen.		
<b>Schlüsselqualifikationen:</b> Lern- und Arbeitstechniken; analytisches Denken; präzises Formulieren.		
<b>Arbeitsaufwand:</b> Gesamt: 120 Std. 45 Std. Vorlesung (Präsenzstudium) 15 Std. Übung (Präsenzstudium) 23 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)		
<b>Voraussetzungen:</b> Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes, insbesondere im Bereich der Datenstrukturen. Modul Informatik 3 (INF-0111) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 0.5 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Constrained data structures (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 3		
<b>Inhalte:</b> Amortization; Self-adjusting data structures: List updates - Splay trees - Pairing heaps; Worst-case-efficient data structures: Deamortization - Global rebuilding - Transformations; Integer data structures: van Emde Boas trees - Fusion trees - Integer priority queues; Geometric data structures: k-d trees - Range trees; Storage-efficient structures: Succinct structures - Algorithms in the read-only model.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Thomas H. Cormen, Charles E. Rivest, Ronald L. Leiserson, Clifford Stein (2009). Introduction to Algorithms (3rd ed.). MIT Press and McGraw-Hill. ISBN 0-262-03384-4.</li> <li>• Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars (2008). Computational Geometry (3rd revised ed.). Springer Verlag. ISBN 3-540-77973-6.</li> <li>• Ausgewählte Originalliteratur</li> </ul>		
<b>Modulteil: Constrained data structures (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 1		

---

**Prüfung**

**Constrained data structures (mündliche Prüfung)**

Mündliche Prüfung, Dauer: 30-45 Minuten

<b>Modul INF-0051: Algorithmen für NP-harte Probleme</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<b>Lernziele/Kompetenzen:</b> Kenntnis verschiedener algorithmischer Lösungsansätze für NP-harte Probleme und die Fähigkeit, diese sinnvoll im Kontext neuer Probleme einzusetzen, zu analysieren und zu bewerten.		
<b>Schlüsselqualifikationen:</b> Lern- und Arbeitstechniken; analytisches Denken; präzises Formulieren.		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b> Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes, insbesondere im Bereich der Graphenalgorithmen. Modul Informatik 3 (INF-0111) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<b>Moduleil: Algorithmen für NP-harte Probleme (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 4		
<b>Inhalte:</b> NP-harte Probleme können nach heutigem Wissen nicht in polynomieller Zeit auf einem üblichen Rechner gelöst werden. Ungeachtet dessen treten solche Probleme überaus häufig in der Praxis auf, z.B. bei vielen Planungsaufgaben, und es ist von großer ökonomischer Bedeutung, sie doch noch zu lösen, zumindest "so gut wie es geht". Die Vorlesung behandelt Methoden der Algorithmentheorie, die hierfür entwickelt wurden. Einige Stichpunkte: Approximationsalgorithmen, Branch-and-Bound, Parametrisierung. Es werden auch Grenzen dieser Methoden aufgezeichnet.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Skript</li> <li>• Ausiello et al., Complexity and Approximation, Springer, Berlin, 1999.</li> <li>• J. Hromkovic, Algorithmics for Hard Problems, Springer, Berlin, 2001.</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b>		
<b>Algorithmen für NP-harte Probleme (Vorlesung)</b> NP-harte Probleme können nach heutigem Wissen nicht in polynomieller Zeit auf einem üblichen Rechner gelöst werden. Ungeachtet dessen treten solche Probleme überaus häufig in der Praxis auf, z.B. bei vielen Planungsaufgaben, und es ist von großer ökonomischer Bedeutung, sie doch noch zu lösen, zumindest "so gut wie es geht". Die Vorlesung behandelt Methoden der Algorithmentheorie, die hierfür entwickelt wurden. Einige Stichpunkte: Approximationsalgorithmen, Branch-and-Bound, Parametrisierung. Es werden auch Grenzen dieser Methoden aufgezeigt.		

**Modulteil: Algorithmen für NP-harte Probleme (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Algorithmen für NP-harte Probleme (Übung)**

**Prüfung**

**Algorithmen für NP-harte Probleme (mündliche Prüfung)**

Mündliche Prüfung, Dauer: 30-45 Minuten

<b>Modul INF-0052: Einführung in die Komplexitätstheorie</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<b>Lernziele/Kompetenzen:</b> Verständnis für zentrale Fragen und Methoden der Komplexitätstheorie und die Fähigkeit, einfache komplexitätstheoretische Fragestellungen zu klären.		
<b>Schlüsselqualifikationen:</b> Lern- und Arbeitstechniken; analytisches Denken; präzises Formulieren.		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b> Empfehlenswert: Gutes Verständnis des Stoffes aus Einführung in die Theoretische Informatik sowie Informatik III, insbesondere bzgl. Turing-Maschinen und Graphenalgorithmen. Modul Informatik 3 (INF-0111) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Einführung in die Komplexitätstheorie (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> Aufbauend auf den in den Grundvorlesungen Einführung in die Theoretische Informatik und Informatik III gelegten Grundlagen werden wichtige Aspekte der Komplexitätstheorie behandelt. Das Anliegen der Komplexitätstheorie ist es, die inhärente Schwierigkeit von Berechnungsproblemen zu untersuchen und somit die prinzipiellen Grenzen effizienter Algorithmen zu beleuchten.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Skript</li> <li>• Christos H. Papadimitriou, Computational Complexity, Addison-Wesley, Reading, Mass., 1994.</li> </ul>		
<b>Modulteil: Einführung in die Komplexitätstheorie (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Prüfung</b> <b>Einführung in die Komplexitätstheorie (mündliche Prüfung)</b> Mündliche Prüfung, Dauer: 30-45 Minuten		



<b>Modul INF-0053: I/O-effiziente Algorithmen</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<b>Lernziele/Kompetenzen:</b> Verständnis für den effizienten Umgang mit Speicherhierarchien; Kenntnis grundlegender I/O-effizienter Algorithmen, insbesondere für Sortieren und verwandte Probleme; Fähigkeit zur Analyse und Bewertung einfacher neuer Algorithmen im I/O-Modell; Verständnis für die Grenzen I/O-effizienter Algorithmen.		
<b>Schlüsselqualifikationen:</b> Lern- und Arbeitstechniken; analytisches Denken; präzises Formulieren.		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)		
<b>Voraussetzungen:</b> Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes. Modul Informatik 3 (INF-0111) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: I/O-effiziente Algorithmen (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Das klassische Berechnungsmodell der Random-Access-Machine (RAM) stößt zunehmend an seine Grenzen. Der Grund ist, dass moderne Rechner nicht über den "flachen" Speicher der RAM verfügen, bei dem alle Speicherzellen "gleichberechtigt" sind, sondern eine ausgefeilte Speicherhierarchie mit Caches, Hauptspeicher und Hintergrundspeicher(n) besitzen. Im Allgemeinen sind "näher am CPU" gelegene Speicher deutlich schneller, dafür aber kleiner, und ein effizienter Algorithmus muss versuchen, häufig benutzte Daten in Speicher mit kurzen Zugriffszeiten zu halten. In der Vorlesung werden wir uns, nach einer Einführung geeigneter Speichermodelle, aus theoretischer Sicht mit sogenannten I/O-effizienten oder "speicherbewussten" Algorithmen befassen, die die Anzahl der Datentransporte zwischen Stufen der Speicherhierarchie möglichst gering halten. Bereits für das Problem des Sortierens wird sich herausstellen, dass die "I/O-effiziente Welt" ganz anders aussieht als die "RAM-Welt".		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Skript</li> <li>• J.S. Vitter, Algorithms and data structures for external memory, Foundations and Trends in Theoretical Computer Science 2 (2008), pp. 305-474</li> </ul>		
<b>Modulteil: I/O-effiziente Algorithmen (Übung)</b>		
<b>Lehrformen:</b> Übung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		

---

**Prüfung**

**I/O-effiziente Algorithmen (mündliche Prüfung)**

Mündliche Prüfung, Dauer: 30-45 Minuten

<b>Modul INF-0054: Datenstrukturen</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<b>Lernziele/Kompetenzen:</b> Kenntnis einer Reihe von nichtelementaren Datenstrukturen und ihrer Analyse; Fähigkeit zur Anpassung dieser Datenstrukturen an neue Anwendungen und zur Entwicklung neuer einfacher Datenstrukturen.		
<b>Schlüsselqualifikationen:</b> Lern- und Arbeitstechniken; analytisches Denken; präzises Formulieren.		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 60 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
<b>Voraussetzungen:</b> Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes. Modul Informatik 3 (INF-0111) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Datenstrukturen (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 4
<b>Inhalte:</b> Datenstrukturen realisieren abstrakte Datentypen so, dass die Operationen der Datentypen besonders effizient ausgeführt werden können. Beispiele von Datenstrukturen sind balancierte Bäume und Hashtabellen. Datenstrukturen können mit objektorientierten Programmiersprachen als Klassen zur Verfügung gestellt werden. In der Vorlesung werden verschiedene Datenstrukturen behandelt, die über die in Informatik III behandelten Datenstrukturen hinausgehen, unter anderem die sogenannten dynamischen Bäume von Sleator und Tarjan, Range-Query-Strukturen und Suffix-Bäume.
<b>Literatur:</b> Skript
<b>Modulteil: Datenstrukturen (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2
<b>Prüfung</b> <b>Datenstrukturen (mündliche Prüfung)</b> Mündliche Prüfung, Dauer: 30-45 Minuten

<b>Modul INF-0055: Teile-und-Herrsche-Algorithmen</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<b>Lernziele/Kompetenzen:</b> Die Fähigkeit, das fundamentale Teile-und-Herrsche-Prinzip mit neuen Ideen zu kombinieren, um so neue Algorithmen zu erhalten; die Studierenden sind in der Lage, Teile-und-Herrsche-Algorithmen zu verstehen und zu analysieren.		
<b>Schlüsselqualifikationen:</b> Lern- und Arbeitstechniken; analytisches Denken; präzises Formulieren.		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)		
<b>Voraussetzungen:</b> Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes, insbesondere im Bereich der Graphalgorithmen. Modul Informatik 3 (INF-0111) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Teile-und-Herrsche-Algorithmen (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> Teile-und-Herrsche-Algorithmen wie Sortieren durch Mischen kennt jeder. Aber wie kann man das Teile-und-Herrsche-Prinzip nutzen, um Probleme wie Vertex Cover und das Closest Points-Problem zu lösen? Die Vorlesung zeigt, wie dieses fundamentale Prinzip mit weiteren Ideen kombiniert werden kann, um so zum Beispiel Probleme aus der algorithmischen Geometrie, der Mathematik und der Graphentheorie zu lösen.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Dasgupta, Papadimitriou, und Vazirani. Algorithms. McGraw-Hill 2006, 2. Kapitel. Güting und Dieker. Datenstrukturen und Algorithmen. Vieweg und Teubner Verlag, 2004, 7. Kapitel.</li> <li>• Boncelet. Block Arithmetic Coding for Source Compression, IEEE Trans. Inform. Theory, IT-39, 1993, Seiten 1546-1554.</li> <li>• Niedermeier. Invitation to Fixed-Parameter Algorithms. Oxford Press 2006, Kapitel 1-5.</li> <li>• Kneis, Mölle, Richter, Rossmanith. Divide-and-Color. WG 2006, LNCS 4271, Seiten 58-67.</li> </ul>		
<b>Modulteil: Teile-und-Herrsche-Algorithmen (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		

---

**Prüfung**

**Teile-und-Herrsche-Algorithmen (mündliche Prüfung)**

Mündliche Prüfung, Dauer: 30-45 Minuten

<b>Modul INF-0056: Online-Algorithmen</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<b>Lernziele/Kompetenzen:</b> Verständnis der Online-Problematik; Kenntnis fundamentaler Online-Probleme und -Algorithmen; Fähigkeit zum selbstständigen Entwurf einfacher Online-Algorithmen, zu ihrer kompetitiven Analyse mittels Potentialfunktionen und zu ihrer Bewertung.		
<b>Schlüsselqualifikationen:</b> Lern- und Arbeitstechniken; analytisches Denken; präzises Formulieren.		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Online-Algorithmen (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Manchmal muss man Entscheidungen treffen, bevor alle relevanten Daten bekannt sind. Will man z. B. Aktien kaufen, so wäre es sehr hilfreich, über die künftige Entwicklung aller Aktienkurse informiert zu sein; aber es liegt in der Natur der Sache, dass man den Kauf tätigen muss, bevor diese Information vorliegt. Ein zweites Beispiel: Eine Funktaxizentrale muss nach jeder Bestellung einen der verfügbaren Wagen auswählen und zum Fahrgast schicken; mit Wissen über später eintreffende Anrufe könnten die Wagen vielleicht günstiger auf die Fahrgäste verteilt werden. Algorithmen, die Entscheidungen bei unvollständiger Information treffen, heißen Online-Algorithmen. Die Vorlesung behandelt Online-Algorithmen und ihre Analyse.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Skript</li> <li>• A. Borodin und R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998.</li> </ul>		
<b>Modulteil: Online-Algorithmen (Übung)</b>		
<b>Lehrformen:</b> Übung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Prüfung</b>		
<b>Online-Algorithmen (mündliche Prüfung)</b> Mündliche Prüfung, Dauer: 30-45 Minuten		

<b>Modul INF-0057: Praktikum: NP-harte Graphprobleme</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<b>Lernziele/Kompetenzen:</b> Programmiererfahrung; die Studierenden sind in der Lage, Algorithmen für NP-harte Graphprobleme aus wissenschaftlichen Veröffentlichungen zu analysieren und einfache Algorithmen, die auftretende Subprobleme lösen, zu entwickeln. <b>Schlüsselqualifikationen:</b> Team- und Kommunikationsfähigkeit; Lern- und Arbeitstechniken; Fähigkeit zur Analyse und Präsentation abstrakter Sachverhalte.		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
<b>Voraussetzungen:</b> Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes, insbesondere im Bereich der Graphalgorithmen. Modul Informatik 3 (INF-0111) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Praktikum: NP-harte Graphprobleme</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 6
<b>Inhalte:</b> In der Informatik III werden einige Probleme als NP-hart klassifiziert. Es wird allgemein erwartet, dass diese Probleme nicht in voller Allgemeinheit in Polynomialzeit gelöst werden können. Ungeachtet dessen sind NP-harte Probleme in der Praxis von großer Bedeutung. Das Ziel des Praktikums ist, neben praktischer Programmiererfahrung einige der in der Informatik III vorgestellten Graphalgorithmen zu implementieren und so zu erweitern, dass komplexere Probleme gelöst werden können. Im Praktikum werden, aufbauend auf den Graphalgorithmen der Informatik III, verschiedenste Algorithmen für NP-harte Graphprobleme in C++ implementiert.
<b>Literatur:</b> Ausgewählte wissenschaftliche Artikel.

<b>Prüfung</b> <b>Abschlussbericht, Präsentation, Softwareabgabe</b> Praktikum
--

<b>Modul INF-0058: Seminar Algorithmen und Datenstrukturen für Master</b>		4 ECTS/LP
Version 1.5.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<b>Lernziele/Kompetenzen:</b> Beherrschung der Grundlagen des wissenschaftlichen Arbeitens; Fähigkeit zu guter schriftlicher und mündlicher Kommunikation wissenschaftlicher Sachverhalte.  <b>Schlüsselqualifikationen:</b> Lern- und Arbeitstechniken; Kommunikationsfähigkeit; Fähigkeit zur Literaturrecherche und zum Einsatz neuer Medien		
<b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)		
<b>Voraussetzungen:</b> Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes.		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Algorithmen und Datenstrukturen</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Aktuelle und klassische Themen aus dem Bereich Algorithmen und Datenstrukturen werden anhand von Originalliteratur behandelt.		
<b>Literatur:</b> Ausgewählte wissenschaftliche Artikel.		
<b>Prüfung</b>		
<b>Schriftliche Ausarbeitung und Vortrag</b> Seminar		



<b>Modul INF-0059: Projektmodul Theoretische Informatik</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet der theoretischen Informatik zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebietes in Forschungsprojekten zu entwickeln. Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Qualitätsbewusstsein, Akribie. Selbständige Arbeit, Zeitmanagement, eigenständige Literaturrecherche, Arbeit mit englischsprachiger Fachliteratur, Grundsätze guter wissenschaftlicher Praxis.</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Projektmodul Theoretische Informatik</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 1</p>		
<p><b>Inhalte:</b> Autonome Mitarbeit an aktuellen Forschungsthemen.</p>		
<b>Prüfung</b>		
<p><b>Vortrag und schriftliche Ausarbeitung</b> Praktikum</p>		

<b>Modul INF-0107: Seminar Petrinetze</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Robert Lorenz		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden sind in der Lage, ein wissenschaftlich anspruchsvolles Thema aus dem Gebiet "Petrinetze" selbstständig zu erarbeiten, dieses klar, verständlich und überzeugend in Schrift und Vortrag zu präsentieren und sachlich über Vorträge zu diskutieren.</p> <p>Sie verfügen über die dafür notwendige wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fertigkeit der Dokumentation und verständlichen, sicheren und überzeugenden Darstellung von Ideen, Konzepten und Ergebnissen; Kommunikationsfähigkeit; Fähigkeit zum Einsatz neuer Medien; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Qualitätsbewußtsein;Wissenschaftliche Methodik;</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<p><b>Voraussetzungen:</b> Modul Halbordnungssemantik paralleler Systeme (INF-0099) - empfohlen Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen Modul Logik für Informatiker (INF-0155) - empfohlen Modul Petrinetze - eine Theorie paralleler Systeme (INF-0161) - empfohlen</p>		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Petrinetze</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<p><b>Inhalte:</b> Aktuelle Forschungsarbeiten zu Konstruktion, Analyse, Simulation, Synthese und Verifikation von Modellen nebenläufiger Systeme mit Petrinetzen, sowie zur Theorie von Petrinetz-Transduktoren und deren Anwendung in der Implementierung von Sprachdialogsystemen.</p> <p>Das Seminar dient ausschließlich zur Vorbereitung auf Abschlussarbeiten und deren Begleitung.</p>		
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Aktuelle Forschungsbeiträge</li> <li>• Projekt-Homepage VipTool: <a href="http://www.fernuni-hagen.de/sttp/forschung/vip_tool.shtml">http://www.fernuni-hagen.de/sttp/forschung/vip_tool.shtml</a></li> <li>• Projekt-Homepage SYNOPS: <a href="http://www.informatik.uni-augsburg.de/lehrstuehle/inf/projekte/synops/">http://www.informatik.uni-augsburg.de/lehrstuehle/inf/projekte/synops/</a></li> </ul>		
<b>Prüfung</b>		
<b>Vortrag und schriftliche Ausarbeitung</b>		
Seminar		

<b>Modul INF-0108: Projektmodul Lehrprofessur für Informatik</b> <i>Project Module Teaching Professorship Informatics</i>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Robert Lorenz		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul haben die Studierenden tiefergehende Fachkenntnisse und Fähigkeiten auf einem der Gebiete "Nebenläufige Systeme" und "Semantische Dialogmodellierung" erworben, die es ihnen ermöglichen, an die internationale Forschung anzuknüpfen. Sie sind fähig, innovative Methoden bei der Lösung von Problemen in diesem Gebiet anzuwenden und einen wissenschaftlichen Beitrag zu diesem Gebiet zu leisten.</p> <p>Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Recherche in englischsprachiger Literatur; Verständliche, sichere und überzeugende Präsentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewußtsein; Kommunikationsfähigkeit; Fertigkeit der Zusammenarbeit in Teams und Verstehen von Teamprozessen; Grundsätze guter wissenschaftlicher Praxis; Projektmanagementfähigkeiten; Wissenschaftliche Methodik;</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
<b>Voraussetzungen:</b> Grundkenntnisse in einschlägigen Forschungsgebieten des Lehrstuhls		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<p><b>Modulteil: Projektmodul Lehrprofessur für Informatik</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch / Englisch <b>SWS:</b> 1</p>		
<p><b>Inhalte:</b> Mitarbeit an der Entwicklung formaler Grundlagen und theoretischer Ergebnisse, dem Entwurf und der Programmierung unterstützender Softwaretools und der Evaluation von Ergebnissen und Konzepten in aktuellen Forschungsprojekten des Lehrstuhls aus den Bereichen "Nebenläufige Systeme" und "Semantische Dialogmodellierung". Mögliche Themen: Synthese von Petrinetzen aus nicht-sequentiellen Verhaltensbeschreibungen, Process Mining Techniken, Entfaltung von Petrinetzen und Entfaltungs-basiertes Model-Checking, Finite State Transducer in der semantischen Dialogmodellierung, Petrinetz-Transduktoren, Hierarchische kognitive dynamische Systeme zur Signalverarbeitung, Dialog-Strategien, Konfiguration von Spracherkennern, Benutzermodelle in der Spracherkennung, Wizard-of-Oz Experimente zur Erstellung lokaler Grammatiken, Unifikationsalgorithmen</p>		

**Literatur:**

- J. Desel, W. Reisig, G. Rozenberg: Lectures on Concurrency and Petri Nets, Springer, Lecture Notes in Computer Science 3098, 2004
- Projekt-Homepage VipTool: [http://www.fernuni-hagen.de/sttp/forschung/vip\\_tool.shtml](http://www.fernuni-hagen.de/sttp/forschung/vip_tool.shtml)
- Projekt-Homepage SYNOPS: <http://www.informatik.uni-augsburg.de/lehrstuehle/inf/projekte/synops/>
- Daniel Jurafsky & James H. Martin: Speech and Language Processing
- M. Huber; C. Kölbl; R. Lorenz; R. Römer; G. Wirsching: Semantische Dialogmodellierung mit gewichteten Merkmal-Werte-Relationen. In: Rüdiger Hoffmann (Hrsg.), Elektronische Sprach-signalverarbeitung 2009, Tagungsband der 20. Konferenz, 2009, Studentexte zur Sprachkommunikation 54, Seiten 25-32
- M. Droste, W. Kuich, H. Vogler (Eds.): Handbook of Weighted Automata. Monographs in Theoretical Computer Science, Springer, 2009.
- A. Esposito (Eds.): Behavioral Cognitive Systems. LNCS 7403, Springer, 2012

**Prüfung**

**Vortrag und Abschlussbericht**

Praktikum

<b>Modul INF-0116: Algebraische Semantik und Algebraische Systementwicklung</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden haben ein vertieftes Verständnis von algebraischen Beschreibungsmethoden für formale Semantiken. Sie wissen, wie diese Methoden auf Programmiersprachen und ihre Logiken sowie auf andere Systemmodelle wie parallele oder hybride Systeme angewendet werden. Außerdem wissen sie, wie die Algebra durch automatische Beweissysteme unterstützt werden kann.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vorlesung (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Übung (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> Modul Diskrete Strukturen für Informatiker (INF-0109) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<p><b>Modulteil: Algebraische Semantik und Algebraische Systementwicklung (Vorlesung)</b>  <b>Lehrformen:</b> Vorlesung  <b>Sprache:</b> Deutsch  <b>SWS:</b> 4</p>
<p><b>Inhalte:</b> Halbringe, Testelemente, Modale Operatoren, Iterationsoperatoren, Terminierungsanalyse, Wissens-/ Glaubenslogiken, Temporale Logiken, Algebra paralleler Systeme.</p>
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Eigenes Skriptum</li> <li>• U. Hebisch, H. J. Weinert: Halbringe - Algebraische Theorie und Anwendungen in der Informatik, Teubner 1993</li> </ul>
<p><b>Zugeordnete Lehrveranstaltungen:</b>  <b>Algebraische Semantik und Algebraische Systementwicklung (Vorlesung)</b>          Eine formale Semantik besteht aus einem mathematischen Modell für eine Programmier- oder Systembeschreibungssprache. Sie dient dazu, gewisse Beziehungen zwischen Programmen/Systemen und/oder ihren Spezifikationen zu beweisen. Dies betrifft zum einen die Verifikation eines Systems relativ zu einer gegebenen Spezifikation, zum anderen die Transformation, d. h. verfeinernde Umformung von Spezifikationen in ausführbare Systeme oder von Systemen in effizientere Formen. Ziel moderner algebraischer Ansätze hierzu ist es, sowohl die in der semantischen Beschreibung involvierten Ausdrücke wie auch den zugehörigen Beweiskalkül möglichst knapp und durchsichtig zu machen, indem Logik soweit wie möglich in</p>

Algebra kompaktifiziert wird. Neben den mathematischen Grundlagen werden in der Vorlesung einige wichtige semantische Konzepte sequentieller und paralleler Systeme und die wesentlichsten Transformationstechniken besprochen. Anwendungsbeispiele stammen aus dem Software- und Hardware-Bereich.  
... (weiter siehe Digicampus)

**Modulteil: Algebraische Semantik und Algebraische Systementwicklung (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Algebraische Semantik und Algebraische Systementwicklung (Übung)**

Übungen zu Algebraische Semantik und Algebraische Systementwicklung

**Prüfung**

**Algebraische Semantik und Algebraische Systementwicklung (Klausur)**

Klausur / Prüfungsdauer: 120 Minuten

<b>Modul INF-0118: Seminar Programmiermethodik und Multimediale Informationssysteme für Master</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "Programmiermethodik und Multimediale Informationssysteme" selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken, Literaturrecherche, Arbeit mit englischer Fachliteratur, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> in der Regel mind. 1x pro Studienjahr	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<p><b>Modulteil:</b> Seminar Programmiermethodik und Multimediale Informationssysteme für Master</p> <p><b>Lehrformen:</b> Seminar</p> <p><b>Sprache:</b> Deutsch</p> <p><b>SWS:</b> 2</p>
<p><b>Inhalte:</b> Themen aus den Bereichen "Theoretische Informatik", "Multimedia" oder "Datenbanken und Informationssysteme"</p>
<p><b>Literatur:</b> wird jeweils bekanntgegeben</p>
<p><b>Zugeordnete Lehrveranstaltungen:</b> Seminar über Theoretische Informatik (Seminar)</p>

<p><b>Prüfung</b> Vortrag und schriftliche Ausarbeitung Seminar</p>
---

<b>Modul INF-0119: Projektmodul Programmiermethodik und Multimediale Informationssysteme</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet "Programmiermethodik und Multimediale Informationssysteme" zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; Bearbeitung konkreter Fallbeispiele; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis; Durchhaltevermögen; Erlernen von Präsentationstechniken; schriftliche Präsentation eigener Ergebnisse</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<p><b>Modulteil: Projektmodul Programmiermethodik und Multimediale Informationssysteme</b>  <b>Lehrformen:</b> Praktikum  <b>Sprache:</b> Deutsch  <b>SWS:</b> 1</p>
<p><b>Inhalte:</b> Anwendung und Erweiterung von Kleene-Algebren, Halbringtheorie und automatisches Beweisen, Datenbanken und Informationssysteme</p>
<p><b>Literatur:</b> aktuelle Forschungspaper</p>

<p><b>Prüfung</b>  <b>Projektabnahme, Vortrag und Abschlussbericht</b>  Praktikum</p>
---



<b>Modul INF-0130: Formale Methoden im Software Engineering</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<b>Lernziele/Kompetenzen:</b> Die Studierenden können formale Methoden für die Programmverifikation, speziell bei sicherheitskritischer Software einsetzen. Sie trainieren die Fertigkeit zum logischen und analytischen Denken. Sie können Spezifikationen von Datenstrukturen erstellen und deren Eigenschaften formal beweisen. Sie sind in der Lage, funktionale Eigenschaften von Programmen zu formulieren und dafür Beweise zu entwickeln. Sie haben die Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen und können mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen abgeben.		
<b>Schlüsselqualifikationen:</b> Training des logischen Denkens, analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Formale Methoden im Software Engineering (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Übergeordnetes Ziel ist die Produktion beweisbar korrekter Software. In der Vorlesung werden verschiedene klassische Methoden für die Programmverifikation im Kleinen behandelt. Darüber hinaus werde innovative Techniken für die formale Modellierung und Verifikation großer Systeme vermittelt. Als Werkzeug kommt das KIV-System zum Einsatz, das die formale Spezifikation und Verifikation von Systemen ermöglicht. Konkrete Inhalte sind: Algebraische Spezifikationen, interaktives Theorembeweisen, Hoare-Logik, Dynamische Logik, Temporallogik		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Sperschneider, Antoniou: Logic: A Foundation for Computer Science, Addison Wesley 1991</li> <li>• Loeckx, Ehrich, Wolf: Specification of Abstract Data Types, Wiley 1996</li> <li>• Ausführliche Dokumentation</li> <li>• Folienhandout</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b>		
<b>Formale Methoden im Software Engineering (Vorlesung)</b> Die Verwendung formaler Methoden bei der Entwicklung korrekter Software steht an der Schwelle der kommerziellen Nutzung. Das KIV-System ist ein Werkzeug, das die formale Spezifikation, Verifikation und		

Synthese von Programmen ermöglicht. Es wird seit mehreren Jahren entwickelt und inzwischen in industriellen Studien erprobt. Übergeordnetes Ziel ist die Produktion beweisbar korrekter Software. Die Lehrveranstaltung vermittelt den "state of the Art" des Einsatzes formaler Methoden bei der Softwareentwicklung. Es werden Spezifikationstechniken zur Beschreibung und Methoden zum Nachweis der Korrektheit von Softwaresystemen behandelt. Die Lehrveranstaltung beginnt mit der (algebraischen) Spezifikation von Datentypen und stellt dann Kalküle und Vorgehensweisen für die Verifikation sequentieller und paralleler Programme vor. Weiterhin wird Refinement als systematisches Entwicklungsvorgehen von einer abstrakten Spezifikation zu einer korrekten Implementierung vorgestellt. Die Rechner im Raum ... (weiter siehe Digicampus)

**Modulteil: Formale Methoden im Software Engineering (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 4

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Formale Methoden im Software Engineering (Übung)**

**Prüfung**

**Formale Methoden im Software Engineering (mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

<b>Modul INF-0156: Algebraische Beschreibung paralleler Prozesse</b>		6 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<b>Lernziele/Kompetenzen:</b> Die Studierenden besitzen die Fähigkeit, verteilte Systeme auf eine exakte, algebraische Weise (nämlich in der Prozessalgebra CCS) zu modellieren. Sie kennen einen Mechanismus, mit dem man in derartigen Ansätzen eine operationale Semantik definieren kann, und sind dadurch in der Lage, auch andere Prozessalgebren anzuwenden. Sie wissen, welche Anforderungen man an Äquivalenzbegriffe stellen muss und können formal prüfen, ob ein System eine, ebenfalls in CCS geschriebene, Spezifikation erfüllt.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fertigkeit zur Analyse und Strukturierung von Informatikproblemstellungen; Qualitätsbewusstsein, Akribie		
<b>Arbeitsaufwand:</b> Gesamt: 180 Std. 15 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium) 23 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 75 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)		
<b>Voraussetzungen:</b> Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen Modul Logik für Informatiker (INF-0155) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 5.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Algebraische Beschreibung paralleler Prozesse (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 3		
<b>Inhalte:</b> Algebraische Spezifikation verteilter Systeme mittels der Prozessalgebra CCS; operationale Semantik mittels SOS-Regeln; Äquivalenz- bzw. Kongruenzbegriffe (starke und schwache Bisimulation, Beobachtungkongruenz); Nachweis von Kongruenzen mittels Axiomen; Einführung in eine Kombination von Bisimulation und Effizienzvergleich		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• R. Milner: Communication and Concurrency, Prentice Hall</li> <li>• L. Aceto, A. Ingolfsdottir, K.G. Larsen, J. Srba: Reactive Systems. Cambridge University Press 2007</li> <li>• J. Bergstra, A. Ponse, S. Smolka (eds.): Handbook of Process Algebras, Elsevier</li> </ul>		
<b>Modulteil: Algebraische Beschreibung paralleler Prozesse (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 1		

**Prüfung**

**Algebraische Beschreibung paralleler Prozesse (mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

<b>Modul INF-0157: Endliche Automaten</b>		5 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme können die Studierenden deterministische Automaten minimieren und das Verfahren mit guter Effizienz automatisieren. Sie haben vertiefte Kenntnisse zur Modellierung von Problemen mit endlichen Automaten und können sich in neue Anwendungen der Automatentheorie einarbeiten. Insbesondere können sie Schaltkreisverhalten und Mealy-Automaten ineinander übersetzen, und sie können mit geeigneten Ergebnissen reguläre von nicht-regulären Sprachen unterscheiden.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Qualitätsbewusstsein, Akribie</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 20 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 48 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 37 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 45 Std. Vorlesung (Präsenzstudium)</p>		
<p><b>Voraussetzungen:</b> Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen Modul Informatik 3 (INF-0111) - empfohlen</p>		
<p><b>Angebotshäufigkeit:</b> unregelmäßig</p>	<p><b>Empfohlenes Fachsemester:</b> ab dem 5.</p>	<p><b>Minimale Dauer des Moduls:</b> 1 Semester</p>
<p><b>SWS:</b> 3</p>	<p><b>Wiederholbarkeit:</b> siehe PO des Studiengangs</p>	
<p><b>Moduleile</b></p>		
<p><b>Modulteil: Endliche Automaten (Vorlesung mit integrierter Übung)</b> <b>Lehrformen:</b> Vorlesung + Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 3</p>		
<p><b>Inhalte:</b> Die Vorlesung vertieft die Kenntnisse über Endliche Automaten aus der Grundvorlesung "Einführung in die theoretische Informatik". Sie behandelt Minimierung, Abschlusseigenschaften und eine Anwendung bei der Lösung diophantischer Gleichungen. Sie stellt Mealy-, Moore- und Büchi-Automaten vor.</p>		
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Hopcroft, (Motwani, Ullman: Introduction to Automata Theory, Languages and Computation; deutsch: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie</li> <li>• Schöning: Theoretische Informatik kurz gefaßt. 5. Auflage</li> <li>• Thomas: Automata on Infinite Objects. Chapter 4 in Handbook of Theoretical Computer Science, Hrsg. van Leeuwen</li> </ul>		
<p><b>Prüfung</b> <b>Endliche Automaten (mündliche Prüfung)</b> Mündliche Prüfung / Prüfungsdauer: 30 Minuten</p>		

<b>Modul INF-0161: Petrinetze - eine Theorie paralleler Systeme</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden werden in die Lage versetzt, parallele bzw. nebenläufige Systeme mit Petrinetzen formal zu modellieren. Anhand verschiedener Verhaltensbegriffe lernen sie die neuartigen Aspekte der Abläufe solcher Systeme kennen. Sie werden befähigt, wichtige Systemeigenschaften mit Petrinetz-spezifischen Methoden nachzuweisen.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fertigkeit zur Analyse und Strukturierung von Informatikproblemstellungen; Qualitätsbewusstsein, Akribie</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<b>Voraussetzungen:</b> Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig (i. d. R. im WS)	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Petrinetze - eine Theorie paralleler Systeme (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Graphenbasierte Modellierung paralleler Systeme mittels verschiedener Varianten von Petrinetzen; verschiedene Verhaltensbeschreibungen (Schalt- und Schrittfolgen, Sprache, Failure-Semantik); Begriffe und Techniken der Verhaltensanalyse (Verklemmung, Lebendigkeit, Fairness; S- und T-Invarianten, Überdeckbarkeitsgraph)		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Desel, Reisig, Rozenberg (eds.): Lectures on Concurrency and Petri Nets. Advances in Petri Nets. Springer, LNCS 3098</li> <li>• Peterson: Petri Net Theory and the Modelling of Systems. Prentice Hall</li> <li>• Reisig: Petrinetze - Eine Einführung. 2. Auflage; Springer</li> </ul>		
<b>Modulteil: Petrinetze - eine Theorie paralleler Systeme (Übung)</b>		
<b>Lehrformen:</b> Übung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Prüfung</b>		
<b>Petrinetze - eine Theorie paralleler Systeme (Mündliche Prüfung)</b> Mündliche Prüfung / Prüfungsdauer: 30 Minuten		

<b>Modul INF-0163: Verteilte Algorithmen</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<b>Lernziele/Kompetenzen:</b> Vertieftes Verständnis für die Probleme und Problemlösungen in verteilten Systemen; Kenntnis wichtiger Algorithmen und ihres Aufwands, Einsicht in ihre Korrektheit; Fähigkeit, solche Algorithmen zu modifizieren sowie zugehörige Korrektheitsbeweise und Aufwandsbestimmungen zu prüfen und selbst zu entwickeln.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fertigkeit zur Analyse und Strukturierung von Informatikproblemstellungen; Kenntnisse der Vor-/Nachteile von Entwurfsalternativen, Bewertung im jeweiligen Anwendungszusammenhang; Qualitätsbewusstsein, Akribie		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)		
<b>Voraussetzungen:</b> Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen Modul Informatik 3 (INF-0111) - empfohlen		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Verteilte Algorithmen (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 4
<b>Inhalte:</b> Algorithmen für Grundprobleme in Netzwerken wie Zugriff auf gemeinsame Ressourcen, Aufbau geeigneter Kommunikationsstrukturen und Konsens; es werden synchrone und asynchrone Netzwerke und Fehlertoleranz betrachtet, der Aufwand bestimmt und Korrektheitsbeweise geführt.
<b>Literatur:</b> Nancy Lynch, Distributed Algorithms. Morgan Kaufmann 1996
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Verteilte Algorithmen (Vorlesung)</b> Diese Vorlesung präsentiert Algorithmen für Grundprobleme in Netzwerken mit zahlreichen Prozessoren, wie sie in den Bereichen Telekommunikation, Prozeßkontrolle und verteilte Informationssysteme auftreten, und untersucht die Effizienz dieser Algorithmen. Grundprobleme sind der Zugriff auf gemeinsame Ressourcen (Problem des gegenseitigen Ausschlusses), Kommunikation (Aufbau geeigneter Strukturen zur Verbreitung von Daten im Netzwerk) und Konsens (z.B. ob eine Transaktion in einem verteilten Datenbanksystem abgebrochen werden muß). Dabei sollen Lösungen fehlertolerant sein, also auch bei fehlerhaftem Verhalten oder Ausfall einiger Prozessoren oder Kommunikationsverbindungen korrekt arbeiten. In der Vorlesung wird die Korrektheit der behandelten Algorithmen in der Regel bewiesen, da sie in vielen Anwendungen (in Telefon-, Reservierungs-, Bank- oder Luftfahrt-Kontroll-Systemen) essentiell ist.

**Modulteil: Verteilte Algorithmen (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Verteilte Algorithmen (Übung)**

Diese Vorlesung präsentiert Algorithmen für Grundprobleme in Netzwerken mit zahlreichen Prozessoren, wie sie in den Bereichen Telekommunikation, Prozeßkontrolle und verteilte Informationssysteme auftreten, und untersucht die Effizienz dieser Algorithmen. Grundprobleme sind der Zugriff auf gemeinsame Ressourcen (Problem des gegenseitigen Ausschlusses), Kommunikation (Aufbau geeigneter Strukturen zur Verbreitung von Daten im Netzwerk) und Konsens (z.B. ob eine Transaktion in einem verteilten Datenbanksystem abgebrochen werden muß). Dabei sollen Lösungen fehlertolerant sein, also auch bei fehlerhaftem Verhalten oder Ausfall einiger Prozessoren oder Kommunikationsverbindungen korrekt arbeiten. In der Vorlesung wird die Korrektheit der behandelten Algorithmen in der Regel bewiesen, da sie in vielen Anwendungen (in Telefon-, Reservierungs-, Bank- oder Luftfahrt-Kontroll-Systemen) essentiell ist.

**Prüfung**

**Verteilte Algorithmen (Mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten



<b>Modul INF-0164: Seminar Theorie verteilter Systeme A</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren und Techniken aus dem Gebiet "Theorie verteilter Systeme" zu verstehen und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit der sicheren und überzeugenden Darstellung von Konzepten und formaler Argumentationen; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit zum logischen, analytischen und konzeptionellen Denken</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Seminar Theorie verteilter Systeme A</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>		
<b>Inhalte:</b> Es werden Arbeiten zu verschiedenen Themen aus dem Bereich "Theorie verteilter Systeme" behandelt.		
<b>Literatur:</b> wird jeweils bekanntgegeben		
<p><b>Prüfung</b> <b>Schriftliche Ausarbeitung</b> Seminar</p>		

<b>Modul INF-0165: Projektmodul Theorie verteilter Systeme</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet "Theorie verteilter Systeme" zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Qualitätsbewusstsein, Akribie</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Projektmodul Theorie verteilter Systeme</b> <b>Sprache:</b> Deutsch <b>SWS:</b> 1</p>		
<b>Inhalte:</b> aktuelle Forschungsthemen in der Theorie verteilter Systeme		
<b>Literatur:</b> wissenschaftliche Papiere, evtl. Handbücher		
<p><b>Prüfung</b> <b>Schriftliche Ausarbeitung</b> Projektarbeit</p>		

<b>Modul INF-0201: Platzeffiziente Algorithmen</b>		5 ECTS/LP
Version 1.0.0 (seit WS15/16) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage, platzeffiziente Algorithmen zu verstehen, zu analysieren und selbst zu entwerfen. Sie verstehen die häufig notwendige Abwägung zwischen Zeit und Platz und kennen wichtige Entwurfsmethoden und grundlegende Datenstrukturen für platzeffiziente Algorithmen ebenso wie eine Anzahl konkreter platzeffizienter Algorithmen.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Erwerb von Abstraktionsfähigkeit, Qualitätsbewusstsein, Akribie.</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p><b>Voraussetzungen:</b> Teilnehmer der Vorlesung sollten über gute algorithmische Kenntnisse verfügen, insbesondere im Bereich Graphenalgorithmen. Modul Informatik 3 (INF-0111) - empfohlen</p>		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Platzeffiziente Algorithmen (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>		
<p><b>Inhalte:</b> Manchmal hat ein Algorithmus eine große Eingabe, aber nur wenig frei beschreibbaren Arbeitsspeicher. Zum Beispiel könnte die Eingabe im Internet für Anfragen zur Verfügung stehen, aber in ihrer Gesamtheit so riesig sein, dass es unmöglich oder unpraktisch ist, sie auf den lokalen Rechner herunterzuladen.  Die Vorlesung beschäftigt sich aus theoretischer Sicht mit Algorithmen, die mit weniger Arbeitsspeicher als klassische Algorithmen für dieselben Probleme auskommen. Der Fokus liegt auf Graphenproblemen wie die Durchführung einer Tiefensuche oder die Berechnung kürzester Wege, aber auch Sortieren und platzeffiziente Datenstrukturen kommen zur Sprache. Ein Großteil der in der Vorlesung vorgestellten Ergebnisse wurde seit 2014 am Lehrstuhl für Theoretische Informatik erzielt. Die Vorlesung behandelt somit ein sehr aktives und aktuelles Forschungsgebiet.</p>		
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Skript</li> </ul>		

---

**Modulteil: Platzeffiziente Algorithmen (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Prüfung**

**Platzeffiziente Algorithmen (mündliche Prüfung)**

Mündliche Prüfung, Dauer: 30-45 Minuten

<b>Modul INF-0243: Process Mining</b> <i>Process Mining</i>		6 ECTS/LP
Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Robert Lorenz		
<p><b>Lernziele/Kompetenzen:</b> Teilnehmer verstehen die folgenden wesentlichen Konzepte der Informatik auf einem wissenschaftlichen Niveau mit ihren mathematisch-formalen Grundlagen: Ereignis-Log, Halbordnung und partielle Sprache, Petrinetz, Nebenläufigkeit, sequentielle und kausale Semantik eines nebenläufigen Systems, Geschäftsprozess. Sie können einfache nebenläufige Systeme in einer geeigneten Modellierungssprache modellieren, sowie verschiedene Verhaltensmodelle zur Analyse und Simulation generieren, bewerten und ineinander überführen. Sie kennen die verschiedenen Process Mining Anwendungsfälle und kennen dazu passende Lösungstechniken mit Qualitätskriterien zu deren Bewertung.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen, konzeptionellen und algorithmischen Denken; Eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Verständliche Präsentation von Ergebnissen; Qualitätsbewußtsein; Grundsätze guter wissenschaftlicher Praxis; Wissenschaftliche Methodik;</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 180 Std. 15 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 75 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		<b>ECTS/LP-Bedingungen:</b> Bestehen der Modulprüfung
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<b>Modulteil: Process Mining (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch / Englisch		
<b>SWS:</b> 3		
<p><b>Inhalte:</b> Die Veranstaltung gibt einen fundierten Überblick über das Forschungsgebiet Process Mining:</p> <ul style="list-style-type: none"> <li>- Modelling Techniques: Workflow Nets, Partial Languages</li> <li>- Discovering algorithms</li> <li>- Conformance Checking</li> <li>- Process Enhancement</li> <li>- The PROM Framework</li> </ul>		

**Literatur:**

Wil M. P. van der Aalst:

Process Mining - Data Science in Action, Second Edition. Springer 2016, ISBN 978-3-662-49850-7

Wil M. P. van der Aalst, Boudewijn F. van Dongen: Discovering Petri Nets from Event Logs. Trans. Petri Nets and Other Models of Concurrency 7: 372-422 (2013)

Robin Bergenthum, Jörg Desel, Robert Lorenz, Sebastian Mauser: Process Mining Based on Regions of Languages. BPM 2007: 375-383

**Zugeordnete Lehrveranstaltungen:**

**Process Mining** (Vorlesung)

Die Veranstaltung gibt einen fundierten Überblick über das Forschungsgebiet Process Mining. Kapitelübersicht: 1:: Motivation / Überblick, 2: Mathematische Grundlagen 3:: Petrinetze 4: Andere Systemmodelle (Workflow-Netze, EPKs, BPMN, ...) 5: Event-Logs 6: Alpha-Algorithmus und Tools (PROM) 7: Herausforderungen und Qualitätsmaße 8: Weitere Algorithmen (Heuristic Miner, Genetic Miner, ...) 9: Anwendung von Synthesetechniken 10: Weiterführende Ziele (Conformance Checking, Enhancement) 11: Process Mining vs. Data Mining

**Modulteil: Process Mining (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch / Englisch

**SWS:** 1

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Process Mining** (Übung)

**Prüfung**

**Process Mining (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

<b>Modul INF-0068: Interactive Simulation</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<b>Lernziele/Kompetenzen:</b> In this course, the students are taught foundational knowledge about interactive simulations. In particular, in-depth apprehension of methods in the fields of modelling & simulation, representation, numerical methods and computer graphics will empower the student to evaluate and to contribute to the design and the programmatic implementation of interactive simulations.		
<b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> letzte Prüfung: SS 2017	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<b>Modulteil: Interactive Simulation (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Englisch <b>SWS:</b> 2		
<b>Inhalte:</b> The basic concept of modelling & simulation is extended by the notion of user interactions. Differences and common features among several academic and industrial examples will be stressed in order to develop a generalised terminology and methodology for interactive simulations. Interactivity translates into one or several users influencing the model and the simulation process, respectively; accordingly, the course revolves around the changes to the simulation model and the emerging dynamics in respect to the computational processes that result from the introduction of user interactions. Interactivity in simulations necessitates the development and the utilisation of real-time rendering techniques (computer graphics), intense efforts towards optimisation, and a clear understanding of acceptable numerical errors due to systematic approximations. In this course, we shed light on the state-of-the-art and discuss current challenges and their potential solutions, for instance in regard to simulation histories or dynamic abstraction.		
<b>Literatur:</b> aktuelle wissenschaftliche Paper		
<b>Modulteil: Interactive Simulation (Übung)</b>		
<b>Lehrformen:</b> Übung <b>Sprache:</b> Englisch <b>SWS:</b> 2		

**Prüfung**

**combined exam: written project report ...**

Hausarbeit/Seminararbeit

**Prüfung**

**... and oral 15 min (combined exam)**

Mündliche Prüfung / Prüfungsdauer: 15 Minuten

**Beschreibung:**

Die Prüfung kann jedes Semester zu Beginn und Ende der vorlesungsfreien Zeit abgelegt werden.



<b>Modul INF-0088: Bayesian Networks</b>		5 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<b>Lernziele/Kompetenzen:</b> The student understands the core principles of Bayesian Networks and can apply them to many real-world problems of all sorts of different domains such as robots, web search, smart agents, automated diagnosis systems, help systems, and medical systems to name a few. Bayesian Networks are one of the most versatile statistical machine learning technique today. The student will understand, apply, analyse, and evaluate problems from the point of view of Bayesian Networks.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 5.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Bayesian Networks (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> <ol style="list-style-type: none"> <li>1. Basics of Probability Theory</li> <li>2. Example: Bayesian Network based Face Detection</li> <li>3. Inference</li> <li>4. Influence Diagrams</li> <li>5. Parameter Learning</li> <li>6. Example: probabilistic Latent Semantic Analysis (pLSA)</li> </ol>		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Richard E. Neapolitan. Learning Bayesian Networks. Prentice Hall Series in Artificial Intelligence, 2004. ISBN 0-13-012534-2</li> <li>• Daphne Koller, Nir Friedman. Probabilistic Graphical Models: Principles and Techniques. The MIT Press, 2009. 978-0262013192</li> </ul>		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Bayesian Networks (Vorlesung)</b>		

**Modulteil: Bayesian Networks (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Bayesian Networks (Übung)**

**Prüfung**

**Bayesian Networks (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

<b>Modul INF-0092: Multimedia II: Machine Learning and Computer Vision</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<b>Lernziele/Kompetenzen:</b> Teilnehmer dieser Vorlesung beherrschen wichtige Konzepte des maschinellen Lernens, der Datenreduktion, der fortgeschrittenen Bildverarbeitung und des maschinellen Sehens und können diese anwenden. <b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Multimedia II: Machine Learning and Computer Vision (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 4
<b>Inhalte:</b> Die Vorlesung gibt einen guten Überblick über alle Aspekte des maschinellen Lernens und der maschinellen Extraktion von Informationen aus Multimediadaten (z.B. "Google Image Search", "Google Goggles"). Die erlernten Konzepte werden in den Übungen anhand von erfolgreichen Beispielen aus der Praxis ausprobiert, geübt, analysiert und bewertet. Zum Ende des Semesters werden fortgeschrittene Themen wie Objektdetektion und Objekterkennung von Gesichtern und Menschen behandelt. Die Inhalte der Vorlesung umfassen: Machine Learning (Decision Tree Learning, Artificial Neural Networks, Bayesian Learning, Discrete Adaboost), Data Reduction (Quantization (K-Means Clustering, Affinity Propagation), Dimensionality Reduction Techniques (PCA, NMF, Random Projection, MDS)) und Image Processing & Computer Vision (Salient Feature Points and Feature Descriptors, Object Detection (Face/Car/People Detection), Object Recognition (Face Recognition), Image Search with pLSA)
<b>Literatur:</b> Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Multimedia II (Vorlesung)</b> This course addresses state-of-the-art computer vision algorithms that let computers see, learn, and understand image and video content. After being taught the required basics in machine learning, students will - accompanied by practical exercises - get to know the most promising techniques. The topics of the course may be summarized as follows: Machine learning Implementation of algorithms using Intel IPP and MKL Image/video processing Media content analysis The learned concepts will be illustrated by successful examples in practice. The accompanying

exercises will contain some hands-on experiences. Towards the end of the course more advanced topics in object detection and object recognition will be addressed.

**Modulteil: Multimedia II: Machine Learning and Computer Vision (Übung)**

**Lehrformen:** Übung

**Sprache:** Deutsch

**SWS:** 2

**Zugeordnete Lehrveranstaltungen:**

**Übung zu Multimedia II (Übung)**

**Prüfung**

**Multimedia II: Machine Learning and Computer Vision (Klausur)**

Klausur / Prüfungsdauer: 120 Minuten

<b>Modul INF-0093: Probabilistic Robotics</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<b>Lernziele/Kompetenzen:</b> This course covers the basics of robot perception and robot motion from a probabilistic point of view. The student is able to understand, apply, analyse, and evaluate problems in robotics from the perspective of probabilistic robotics. This is currently the most successful and modern approach in robotics with impressive performance under uncertainty.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Probabilistic Robotics (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> <ol style="list-style-type: none"> <li>1. Introduction to Probabilistic Robotics</li> <li>2. Recursive State Estimation</li> <li>3. Gaussian Filters</li> <li>4. Nonparametric Filters</li> <li>5. Robot Motion</li> <li>6. Robot Perception</li> <li>7. Mobile Robot Localization: Markov and Gaussian</li> <li>8. Mobile Robot Localization: Grid and Monte Carlo</li> <li>9. Occupancy Grid Mapping</li> <li>10. SLAM</li> </ol>		
<b>Literatur:</b> Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic Robotics. Springer Verlag.		
<b>Modulteil: Probabilistic Robotics (Übung)</b>		
<b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		

**Prüfung**

**Probabilistic Robotics (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

<b>Modul INF-0094: Maschinelles Lernen</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<b>Lernziele/Kompetenzen:</b> Teilnehmer dieser Veranstaltung verstehen mathematische Grundlagen und Techniken des maschinellen Lernens wie neuronaler Netze und Support Vector Maschinen. Sie können diese analysieren und selbständig auf neue Probleme anwenden.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b>		
keine		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Maschinelles Lernen (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2
<b>Inhalte:</b> Maschinelles Lernen wird heutzutage in vielen praktischen Anwendungen benutzt wie in der Roboternavigation, der Klassifizierung von Spam-E-mails oder der Spracherkennung. Maschinelles Lernen steht für das automatische Lernen des Computers aus Erfahrungen bzw. anhand von Beispielen. Es werden hierbei Muster in den Daten erkannt, anhand derer dann verallgemeinert werden kann, um neue, unbekannte Beispiele klassifizieren zu können. In dieser Vorlesung wird eine Einführung in die mathematischen Grundlagen und Techniken des maschinellen Lernens wie beispielsweise Neuronale Netze und Support Vektor Maschinen gegeben, so dass diese verstanden, analysiert und selbständig auf neue Probleme angewendet werden können. Die behandelten Themen umfassen Wahrscheinlichkeitsverteilungen, Lineare Modelle für Regression und Klassifikation, Neuronale Netze, Kernel Methoden, Sparse Kernel Maschinen und das Kombinieren von Modellen.
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, Berlin, ISBN-13: 978-0387310732</li> </ul>
<b>Modulteil: Maschinelles Lernen (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2

---

**Prüfung**

**Maschinelles Lernen (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten



<b>Modul INF-0095: Seminar Multimedia Computing (MA)</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien aus dem Gebiet des Multimedia Computings und Computer Vision (z.B. Bildverarbeitung, Videoverarbeitung, maschinelles Sehen/Hören und Lernen, Bild-/Videosuche) selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten.</p> <p>Sie verfügen über die wissenschaftliche Methodik, die Kommunikationsfähigkeit und die Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken, Literaturrecherche, Arbeit mit englischer Fachliteratur, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Multimedia Computing (MA)</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Das konkrete Thema des Seminars aus dem weitläufigen Gebiet des Multimedia wird jedes Jahr neu festgelegt und an aktuelle Themen angepasst.		
<b>Literatur:</b> aktuelle Forschungsliteratur		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Seminar über Multimedia und Maschinelles Sehen (Master) (Seminar)</b>		
<b>Prüfung</b> <b>Vortrag und schriftliche Ausarbeitung</b> Seminar		

<b>Modul INF-0096: Projektmodul Multimedia Computing</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität aus dem Gebiet des Multimedia Computings (z.B. Bild-, Video- und Tonverarbeitung bzw. Bild-, Video- und Tonsuche) und des maschinellen Sehens (Objekterkennung, Personendetektion, Posenschätzung von Menschen) zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln.</p> <p>Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten, Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<p><b>Modulteil: Projektmodul Multimedia Computing</b>  <b>Lehrformen:</b> Praktikum  <b>Sprache:</b> Deutsch  <b>SWS:</b> 1</p>
<p><b>Inhalte:</b> Die konkrete Aufgabenstellung aus dem weitenläufigen Gebiet des Multimedia und maschinellen Sehens (Bild-, Video- und Tonverarbeitung, Objekterkennung, Suche in Bild-, Video- und Tonmaterial) wird jedes Jahr aktuell für jeden Studenten einzeln neu entworfen.</p>
<p><b>Literatur:</b> Literaturhinweise werden zum Beginn der Veranstaltung bekanntgegeben.</p>

<p><b>Prüfung</b>  <b>Vortrag mit Softwarepräsentation; Ausarbeitung mit Softwaredokumentation; Erklärung des Quellcodes (Code Review)</b>  Praktikum</p>
---

<b>Modul INF-0112: Graphikprogrammierung</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden haben ein vertieftes Verständnis der wesentlichen Grundlagentechniken für die Erstellung dreidimensionaler Bilder und Animationen. Sie haben zentrale Teile der vorgestellten Verfahren eigenständig programmiertechnisch umgesetzt und können diese in konkreten Fragestellungen anwenden.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; Bearbeitung konkreter Fallbeispiele; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)</p>		
<p><b>Voraussetzungen:</b> Mathematik für Informatiker I + II (alternativ Analysis I + Lineare Algebra I) empfohlen Modul Informatik 1 (INF-0097) - empfohlen Modul Informatik 2 (INF-0098) - empfohlen</p>		
<p><b>Angebotshäufigkeit:</b> unregelmäßig</p>	<p><b>Empfohlenes Fachsemester:</b> ab dem 4.</p>	<p><b>Minimale Dauer des Moduls:</b> 1 Semester</p>
<p><b>SWS:</b> 6</p>	<p><b>Wiederholbarkeit:</b> siehe PO des Studiengangs</p>	
<p><b>Modulteile</b></p>		
<p><b>Modulteil: Graphikprogrammierung (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 4</p>		
<p><b>Inhalte:</b> Koordinaten und Transformationen, Projektionen und Kameramodelle, Sichtbarkeit, Farbmodelle, Beleuchtung und Schattierung, Texturen, Schattenberechnung, Raytracing, Animationstechniken, OpenGL/JOGL</p>		
<p><b>Literatur:</b></p> <ul style="list-style-type: none"> <li>• Eigenes Skriptum</li> <li>• M. Bender, M. Brill, Computergrafik - ein anwendungsorientiertes Lehrbuch, Hanser 2006</li> <li>• F. Hill, S. Kelley: Computer graphics using OpenGL, Pearson 2007</li> </ul>		
<p><b>Modulteil: Graphikprogrammierung (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2</p>		

---

**Prüfung**

**Graphikprogrammierung (Klausur)**

Klausur / Prüfungsdauer: 120 Minuten

<b>Modul INF-0118: Seminar Programmiermethodik und Multimediale Informationssysteme für Master</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "Programmiermethodik und Multimediale Informationssysteme" selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken, Literaturrecherche, Arbeit mit englischer Fachliteratur, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> in der Regel mind. 1x pro Studienjahr	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Moduleile</b>
<p><b>Moduleil:</b> Seminar Programmiermethodik und Multimediale Informationssysteme für Master</p> <p><b>Lehrformen:</b> Seminar</p> <p><b>Sprache:</b> Deutsch</p> <p><b>SWS:</b> 2</p>
<p><b>Inhalte:</b> Themen aus den Bereichen "Theoretische Informatik", "Multimedia" oder "Datenbanken und Informationssysteme"</p>
<p><b>Literatur:</b> wird jeweils bekanntgegeben</p>
<p><b>Zugeordnete Lehrveranstaltungen:</b> Seminar über Theoretische Informatik (Seminar)</p>

<p><b>Prüfung</b> Vortrag und schriftliche Ausarbeitung Seminar</p>
---

<b>Modul INF-0119: Projektmodul Programmiermethodik und Multimediale Informationssysteme</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet "Programmiermethodik und Multimediale Informationssysteme" zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; Bearbeitung konkreter Fallbeispiele; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis; Durchhaltevermögen; Erlernen von Präsentationstechniken; schriftliche Präsentation eigener Ergebnisse</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<p><b>Modulteil: Projektmodul Programmiermethodik und Multimediale Informationssysteme</b>  <b>Lehrformen:</b> Praktikum  <b>Sprache:</b> Deutsch  <b>SWS:</b> 1</p>
<p><b>Inhalte:</b> Anwendung und Erweiterung von Kleene-Algebren, Halbringtheorie und automatisches Beweisen, Datenbanken und Informationssysteme</p>
<p><b>Literatur:</b> aktuelle Forschungspaper</p>

<p><b>Prüfung</b>  <b>Projektabnahme, Vortrag und Abschlussbericht</b>  Praktikum</p>
---

<b>Modul INF-0170: Projektmodul Human-Centered Multimedia</b>		10 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p><b>Lernziele/Kompetenzen:</b> Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet "Human-Centered Multimedia" zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten; Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen; Verstehen von Teamprozessen; Fertigkeit der Zusammenarbeit in Teams; Fähigkeit zur Leitung von Teams; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Fähigkeit, Beiträge zur Wissenschaft zu leisten; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen; Qualitätsbewusstsein, Akribie</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Projektmodul Human-Centered Multimedia</b>		
<b>Lehrformen:</b> Praktikum		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 1		
<b>Inhalte:</b> Mitarbeit an aktuellen Forschungsthemen.		
<b>Literatur:</b> Literaturhinweise werden je nach Thema zu Beginn des Moduls gegeben.		
<b>Prüfung</b>		
<b>Projektabnahme und Vortrag</b> Praktikum		

<b>Modul INF-0175: Multimedia I: Usability Engineering</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage, einschlägige Methoden und Werkzeuge des nutzerzentrierten Designprozesses angemessen zu bewerten und bei der Entwicklung von Softwareprodukten passend einzusetzen.  <b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten.		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 30 Std. Übung (Präsenzstudium) 60 Std. Vorlesung (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b> Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia Grundlagen II (INF-0166) - empfohlen		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Multimedia I: Usability Engineering (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 4		
<b>Inhalte:</b> Methoden, Werkzeuge und Vorgehensweisen zur Gestaltung von gebrauchstauglichen Softwareprodukten		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Ben Shneiderman, "Designing the User Interface: Strategies for Effective Human-Computer Interaction"</li> <li>• Jakob Nielsen, "Usability Engineering"</li> <li>• Helen Sharp, Yvonne Rogers und Jenny Preece, "Interaction Design beyond Human Computer Interaction"</li> </ul>		
<b>Modulteil: Multimedia I: Usability Engineering (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Prüfung</b> <b>schriftliche Abgaben</b> Übung + Praktikum		



<b>Modul INF-0176: Digital Signal Processing II</b>		6 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: PD Dr. Jonghwa Kim		
<b>Lernziele/Kompetenzen:</b> Die Studierenden verfügen über Kenntnisse zu Filterbanken, Analysemethoden stochastischer Signale, zur Funktionsweise von Wavelets und Signalkompression und sind in der Lage, Digitalfilter zu entwerfen, moderne Signalverarbeitungstechniken zu verstehen sowie die erworbenen theoretischen Kenntnisse auf Multimedia-Daten in MATLAB praktisch anzuwenden.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 180 Std. 60 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> wird nicht mehr angeboten!	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

**Modulteile****Modulteil: Digital Signal Processing II (Vorlesung)****Lehrformen:** Vorlesung**Sprache:** Deutsch**SWS:** 4**Inhalte:**

Ziel des Moduls ist es, die in der Vorlesung "Digital Signal Processing I" gewonnenen Grundkenntnisse digitaler Signalverarbeitung zu erweitern. Die Vorlesung beginnt mit Zusammenfassung des in der Vorlesung Digital Signal Processing I behandelten Stoffs und bietet eine erweiterte Einführung in folgende Themenbereiche: z-Transformation, Systemfunktion, FIR-/IIR-Filter, Wavelet-Transformation, Subband Coding, Signalverarbeitung für Mustererkennung und Multimedia-Anwendungen. Die Vorlesung wird ergänzt durch integrierte MATLAB-Übungen.

**Literatur:**

- Alan V. Oppenheim and Roland W. Schaffer, "Discrete-Time Signal Processing", Prentice Hall
- K. Mitra, "Digital Signal Processing: A Computer-Based Approach", McGraw-Hill
- Stéphane Mallat, "A Wavelet Tour of Signal Processing", Academic Press

**Prüfung****Digital Signal Processing II (Klausur)**

Klausur / Prüfungsdauer: 100 Minuten

<b>Modul INF-0177: Einführung in die Künstliche Intelligenz</b>		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<b>Lernziele/Kompetenzen:</b> Die Studierenden beherrschen basale theoretische und praktische Konzepte aus dem Bereich der künstlichen Intelligenz. Weiterhin sind sie nach Vorlesungsteilnahme in der Lage intelligente Verfahren zu nutzen, zu entwickeln und dabei dem Problem adäquate Methoden einzusetzen.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> wird nicht mehr angeboten!	<b>Empfohlenes Fachsemester:</b> 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Einführung in die Künstliche Intelligenz (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2
<b>Inhalte:</b> Einführung, Problemlösen mit Suche und Constraint Satisfaction, Wissensrepräsentation und Reasoning, Räumliches und Zeitliches Schliessen, Planen, Reasoning und Planen mit Unsicherheit, Intelligente Anwendungen.
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• S. Russell&amp;P. Norvig: Artificial Intelligence - A Modern Approach, 3rd Edition, 2010</li> <li>• weitere Literatur wird in der Vorlesung angegeben.</li> </ul>

<b>Modulteil: Einführung in die Künstliche Intelligenz (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2
--

<b>Prüfung</b> <b>Klausur Einführung in die Künstliche Intelligenz</b> Klausur
--

<b>Modul INF-0178: Praktikum Usability Engineering</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden sind mit Methoden und Techniken des Usability Engineering vertraut. Sie sind in der Lage, in kleinen Teams größere Projektaufgaben zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und als Team zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten; Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen; Verstehen von Teamprozessen; Fertigkeit der Zusammenarbeit in Teams; Fähigkeit zur Leitung von Teams; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Fähigkeit, Beiträge zur Wissenschaft zu leisten; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen; Qualitätsbewusstsein, Akribie</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 90 Std. Praktikum (Präsenzstudium)</p>		
<b>Voraussetzungen:</b> Modul Multimedia I: Usability Engineering (INF-0175) - empfohlen		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: Praktikum Usability Engineering</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 6</p>		
<b>Inhalte:</b> Die konkrete Aufgabenstellung für Studentenprojekte wird jedes Jahr neu entworfen.		
<b>Literatur:</b> Literaturhinweise werden je nach Thema zu Beginn des Semesters bekanntgegeben.		
<p><b>Zugeordnete Lehrveranstaltungen:</b> <b>Praktikum Usability Engineering (Praktikum)</b> In diesem Praktikum wird der menschenzentrierte Design Prozess angewandt, um ein interaktives System zu designen, implementieren, evaluieren und analysieren (User-Centered Prototyping). Das theoretische Wissen wurde dazu in der Vorlesung "Multimedia 1: Usability Engineering" erworben. Das "Praktikum Usability Engineering" wird in der Regel im Sommersemester im Anschluss an diese Vorlesung mit wechselnden konkreten Themen angeboten. In diesem Semester werden folgende Themen angeboten: - Ernährungserfassung (Andreas Seiderer) - Komplexe Kommunikationshilfe (Jan-Oliver Wülfing) - Smart Home, Hausautomation (Chi Tai Dang, Andreas Seiderer) - Mobiles Annotationstool (Simon Flutura) - Empfehlungssystem für Energiespartipps (Stephan Hammer) Eine genauere Beschreibung der Themen könnt ihr auf folgender Website finden: <a href="https://www.informatik.uni-augsburg.de/lehrstuehle/hcm/lectures/2017ss/pue/">https://www.informatik.uni-augsburg.de/lehrstuehle/hcm/lectures/2017ss/pue/</a> ... (weiter siehe Digicampus)</p>		

**Prüfung**

**Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation**

Projektarbeit

<b>Modul INF-0179: Einführung in die Spieleprogrammierung</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage, einschlägige Methoden und Prinzipien der Spieleprogrammierung zu bewerten sowie Komponenten, die diese Prinzipien umsetzen, selbstständig zu entwickeln und technisch umzusetzen.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen, Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten.</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<b>Voraussetzungen:</b> Ferienaufgabe		
<b>Angebotshäufigkeit:</b> jedes Sommersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Einführung in die Spieleprogrammierung (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Game Engines, Entscheidungsfindung für KI-Charaktere, Wegfindung und Navigation, Gruppenverhalten und Gruppendynamik, Animationen, Physik, Storytelling, Ein-/Ausgabemethodik, Audio, Game Design, Grafik und Shaderprogrammierung		
<b>Literatur:</b> Skript		
<b>Zugeordnete Lehrveranstaltungen:</b>		
<b>Einführung in die Spieleprogrammierung (Vorlesung)</b> Veranstaltungsseite: <a href="http://www.informatik.uni-augsburg.de/lehrestuehle/hcm/lectures/2017ss/sp/">http://www.informatik.uni-augsburg.de/lehrestuehle/hcm/lectures/2017ss/sp/</a> Über die erfolgreiche Zulassung zur Veranstaltung entscheidet die Bearbeitung der Ferienaufgabe!		
<b>Modulteil: Einführung in die Spieleprogrammierung (Übung)</b>		
<b>Lehrformen:</b> Übung		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 4		
<b>Zugeordnete Lehrveranstaltungen:</b>		
<b>Einführung in die Spieleprogrammierung (Übung) (Übung)</b>		

Veranstaltungsseite: <http://www.informatik.uni-augsburg.de/lehrstuehle/hcm/lectures/2016ss/sp/> Über die erfolgreiche Zulassung zur Veranstaltung entscheidet die Bearbeitung der Ferienaufgabe! Anmeldung nur über die Seite der Vorlesung!

**Prüfung**

**Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation, Übungsaufgaben**

Projektarbeit

<b>Modul INF-0180: Computational Intelligence</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: PD Dr. Jonghwa Kim		
<b>Lernziele/Kompetenzen:</b> Die Vorlesung soll einen umfassenden Überblick über grundlegende Konzepte und verschiedene Methoden der Computational Intelligence (CI) geben, wobei auch verwandte Fachgebiete wie Künstliche Intelligenz, Digital Signal Processing und Machine Learning in den Überblick einbezogen werden. In den Übungen werden ausgewählte CI-Methoden durch eine Projektarbeit in den Bereichen Optimierung und Klassifikation besonders vertieft.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Computational Intelligence (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch / Englisch <b>SWS:</b> 2		
<b>Inhalte:</b> Zu Beginn führt die Vorlesung in das Thema Computational Intelligence (CI) ein. Hierzu werden die Einsatzmöglichkeiten der CI im Vergleich zu klassischen Lösungsansätzen erläutert. Der Hauptteil der Vorlesung behandelt grundlegende Konzepte und Eigenschaften der drei wichtigsten CI-Methoden Evolutionäre Algorithmen, Künstliche Neuronale Netze und Fuzzy Systeme. Dabei werden auch grundlegende Fragen über Möglichkeiten und Grenzen der CI seminaristisch diskutiert. In den Übungen werden zentrale Anwendungsfelder und relevante Tools exemplarisch dargestellt und projektorientierte Versuche zur Klassifikation und Optimierung mit speziellen Tools durchgeführt.		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Andries Engelbrecht, "Computational Intelligence: An Introduction", Wiley &amp; Sons., 2007</li> <li>• Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification", Wiley, 2001</li> <li>• Kruse R., Borgelt C., Klawonn F., Moewes, C., Ruß G., Steinbrecher M., "Com-putational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze", Vieweg+Teubner Verlag, 2012</li> </ul>		
<b>Modulteil: Computational Intelligence (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch / Englisch <b>SWS:</b> 4		

**Prüfung**

**Computational Intelligence (mündliche Prüfung und Projektabnahme)**

Mündliche Prüfung



<b>Modul INF-0181: Praktikum Multimodal Interaction</b> <i>Multimodal Interaction</i>		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André Damian, Ionut, Dipl.-Inf.		
<b>Lernziele/Kompetenzen:</b> Die Studenten sind mit Methoden und Techniken aus dem Bereich "Multimodale Interaction" vertraut. Sie sind in der Lage, in kleinen Teams größere Projektaufgaben (Entwicklung von Softwaremodulen) zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und als Team zu präsentieren. <b>Schlüsselqualifikationen:</b> Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten; Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen; Verstehen von Teamprozessen; Fertigkeit der Zusammenarbeit in Teams; Fähigkeit zur Leitung von Teams; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Fähigkeit Beiträge zur Wissenschaft zu leisten; Kompetenz zum Erkennen von bedeutend-technischen Entwicklungen; Qualitätsbewusstsein, Akribie.		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
<b>Voraussetzungen:</b> Programmiererfahrung		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Praktikum Multimodal Interaction</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 6		
<b>Inhalte:</b> Die konkrete Aufgabenstellung aus dem Gebiet "Multimodal Interaction" wird jedes Semester neu entworfen.		
<b>Literatur:</b> Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Praktikum Multimodal Interaction</b> (Praktikum) In dem Praktikum Multimodal Interaction (früher "Multimodal User Interfaces") werden die Teilnehmer in kleinen Gruppen eigenständig Software entwickeln die mit Hilfe moderner Sensoren und maschinelles Lernen auf einer innovativen Art und Weise auf den Nutzer reagieren kann. Die genaue Aufgabenstellung hängt von der Anzahl an Teilnehmern ab und wird in der ersten Veranstaltung genauer definiert Vorkenntnisse: C++ Kenntnisse sind erforderlich This lecture is offered both in German and in English. Homepage: <a href="https://www.informatik.uni-augsburg.de/lehrstuehle/hcm/lectures/2017ss/mui/">https://www.informatik.uni-augsburg.de/lehrstuehle/hcm/lectures/2017ss/mui/</a> SVN:		

**Prüfung**

**Vortrag mit Softwarepräsentation, Ausarbeitung mit Softwaredokumentation**

Projektarbeit

<b>Modul INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p><b>Lernziele/Kompetenzen:</b> Die Studenten sind mit Methoden und Techniken aus dem Gebiet "Multimodale Echtzeitsignalverarbeitung" vertraut. Sie sind in der Lage, in kleinen Teams größere Projektaufgaben (Entwicklung von Softwaremodulen) zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und als Team zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten; Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen; Verstehen von Teamprozessen; Fertigkeit der Zusammenarbeit in Teams; Fähigkeit zur Leitung von Teams; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Fähigkeit, Beiträge zur Wissenschaft zu leisten; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen; Qualitätsbewusstsein, Akribie</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 90 Std. Praktikum (Präsenzstudium)</p>		
<p><b>Voraussetzungen:</b> Programmiererfahrung  Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia Grundlagen II (INF-0166) - empfohlen</p>		
<p><b>Angebotshäufigkeit:</b> jedes Sommersemester</p>	<p><b>Empfohlenes Fachsemester:</b> ab dem 1.</p>	<p><b>Minimale Dauer des Moduls:</b> 1 Semester</p>
<p><b>SWS:</b> 6</p>	<p><b>Wiederholbarkeit:</b> siehe PO des Studiengangs</p>	

<b>Modulteile</b>
<p><b>Modulteil: Praktikum Multimodale Echtzeitsignalverarbeitung</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 6</p>
<p><b>Inhalte:</b> Die konkrete Aufgabenstellung aus dem weiten Gebiet der multimodalen Echtzeitsignalverarbeitung wird jedes Jahr neu entworfen.</p>
<p><b>Literatur:</b> Literaturhinweise werden je nach Thema zu Beginn des Semesters bekanntgegeben.</p>
<p><b>Zugeordnete Lehrveranstaltungen:</b> <b>Multimodale Echtzeitsignalverarbeitung (Praktikum)</b> Social Signal Interpretation (SSI) ist ein an unserem Lehrstuhl entwickelte Toolbox zur parallelen und synchronen Verarbeitung von Sensor Daten in Echtzeit. Es hilft bei der Entwicklung multimodaler Applikationen, wobei es die Verbindung zu den angeschlossenen Sensoren herstellt und nötige Vorverarbeitung, wie Filtering oder Featureberechnung, bis hin zum Training und Einsatz von Klassifikatoren ermöglicht. In dem Praktikum werden verschiedene Aufgabenstellungen angeboten, die jeweils in kleinen Gruppen bearbeitet werden und die Funktionalität der Toolbox erweitern sollen. Die nötigen Grundkenntnisse zur Bearbeitung des Praktikums werden zu Beginn des Praktikums vermittelt: dies beinhaltet eine Einführung in digitale Signalverarbeitung, die</p>

SSI Toolbox und Python/C++ Programmierung. Bitte beachten: Die Zulassung erfordert die Bearbeitung einer Ferienaufgabe bis zum 21.04.17 . Alle zugelassenen Teilnehmer werden gebeten sich am Montag den 24.04.17 um 10 Uhr in Raum 2026 einzufinden.  
... (weiter siehe Digicampus)

**Prüfung**

**Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation**

Projektarbeit

<b>Modul INF-0183: Praktikum Spieleprogrammierung</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden sind mit Methoden und Prinzipien aus der Spieleprogrammierung vertraut. Sie sind in der Lage, in kleinen Teams größere Projektaufgaben (Entwicklung von Softwaremodulen) zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten; Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen; Verstehen von Teamprozessen; Fertigkeit der Zusammenarbeit in Teams; Fähigkeit zur Leitung von Teams; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Fähigkeit, Beiträge zur Wissenschaft zu leisten; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen; Qualitätsbewusstsein, Akribie</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<b>Voraussetzungen:</b> Modul Einführung in die Spieleprogrammierung (INF-0179) - empfohlen		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 2.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<p><b>Modulteil: <a href="#">Praktikum Spieleprogrammierung</a></b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 6</p>		
<p><b>Inhalte:</b> Innerhalb des Praktikums soll ein Spiel entwickelt werden. Der inhaltliche Schwerpunkt des Praktikums wird jedes Jahr neu festgelegt.</p>		
<p><b>Literatur:</b> Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.</p>		
<p><b>Prüfung</b> <b>Vortrag mit Softwarepräsentation, Ausarbeitung mit Softwaredokumentation</b> Projektarbeit</p>		

<b>Modul INF-0184: Seminar User Interface Design</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "User Interface Design" selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten.		
<b>Schlüsselqualifikationen:</b> Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten;		
<b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar User Interface Design</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Themen aus dem Bereich "User Interface Design"		
<b>Literatur:</b> Literaturhinweise werden bei der Vorbesprechung bekanntgegeben.		
<b>Prüfung</b>		
<b>Vortrag und schriftliche Ausarbeitung</b> Seminar		

<b>Modul INF-0185: Seminar Advanced Topics in Signal and Pattern Recognition</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: PD Dr. Jonghwa Kim		
<p><b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "Advanced Signal and Pattern Recognition" selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten.</p> <p>Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p><b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken, Literaturrecherche, Arbeit mit englischer Fachliteratur, Grundsätze guter wissenschaftlicher Praxis</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> wird nicht mehr angeboten!	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Advanced Topics in Signal and Pattern Recognition</b>		
<b>Lehrformen:</b> Seminar		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 2		
<b>Inhalte:</b> Der Themenbereich für dieses Seminar wird jährlich unter Berücksichtigung neuer Trends in der Signalanalyse und Mustererkennung neu festgelegt.		
<b>Literatur:</b> aktuelle Forschungsliteratur		
<b>Prüfung</b>		
<b>Vortrag und schriftliche Ausarbeitung</b> Seminar		

<b>Modul INF-0198: Intelligente Systeme</b>		5 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<b>Lernziele/Kompetenzen:</b> Die Studierenden beherrschen basale theoretische und praktische Konzepte aus dem Bereich der künstlichen Intelligenz. Weiterhin sind sie nach Vorlesungsteilnahme in der Lage, intelligente Verfahren zu nutzen, zu entwickeln und dabei dem Problem adäquate Methoden einzusetzen.		
<b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
<b>Arbeitsaufwand:</b> Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)		
<b>Voraussetzungen:</b> Sinnvoll sind Kenntnisse und Interesse an: Diskreten Strukturen und Logiken		
<b>Angebotshäufigkeit:</b> unregelmäßig	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 4	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Intelligente Systeme (Vorlesung)</b>		
<b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Inhalte:</b> <ul style="list-style-type: none"> <li>• Wissensrepräsentation, Sprachen, Ontologien (Prädikatenlogik, Hornlogik, Semantische Netze, ...)</li> <li>• Der Inferenzbegriff ("Beweis"kalküle, Resolution)</li> <li>• Klassifikation und Konzepte (Clustering, Hierarchical Clustering, Incremental Clustering)</li> <li>• Konzeptanalyse und Konzeptionshierarchien (Formal Concept Analysis)</li> <li>• Relationale und Soft-Computing Ansätze zur Datenanalyse (Grobmengen, Fuzzy-Logik, ...)</li> <li>• Generalisierung und Induktion (Subsumption, Negation, Abduktion, ...)</li> <li>• Lernen: Finden von Hypothesen (Induktives Logisches Programmieren, Bagging)</li> </ul>		
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Eigenes Skriptum</li> <li>• S. Russell &amp; P. Norvig: Artificial Intelligence - A Modern Approach, 3rd Edition, 2010</li> <li>• weitere Literatur wird in der Vorlesung angegeben</li> </ul>		
<b>Modulteil: Intelligente Systeme (Übung)</b>		
<b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 2		
<b>Prüfung</b>		
<b>Intelligente Systeme (Klausur)</b> Klausur / Prüfungsdauer: 90 Minuten		



<b>Modul INF-0207: Reinforcement Learning</b>		8 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<b>Lernziele/Kompetenzen:</b> Die Studierenden sind in der Lage, Methoden und Prinzipien des Reinforcement Learning zu bewerten sowie Komponenten, die diese Methoden und Prinzipien umsetzen, selbstständig zu entwickeln und technisch umzusetzen. <b>Schlüsselqualifikationen:</b> Fertigkeit zum logischen, analytischen und konzeptionellen Denken.		
<b>Arbeitsaufwand:</b> Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Übung (Präsenzstudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Wintersemester	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 6	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<b>Modulteile</b>
<b>Modulteil: Reinforcement Learning (Vorlesung)</b> <b>Lehrformen:</b> Vorlesung <b>Sprache:</b> Deutsch <b>SWS:</b> 2
<b>Inhalte:</b> Markow-Entscheidungsproblem, Dynamische Programmierung, Monte Carlo Methoden, TD-Lernen, Eligibility Traces, Hierarchisches Bestärkendes Lernen, Planen und Lernen, Generalisierung
<b>Literatur:</b> <ul style="list-style-type: none"> <li>• Skript</li> <li>• Richard S. Sutton und Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998</li> </ul>
<b>Modulteil: Reinforcement Learning (Übung)</b> <b>Lehrformen:</b> Übung <b>Sprache:</b> Deutsch <b>SWS:</b> 4

<b>Prüfung</b> <b>Reinforcement Learning (Projektarbeit)</b> Projektarbeit, Projektarbeit / mündliche Prüfung / Prüfungsdauer: 30 Minuten
---

<b>Modul INF-0250: Praktikum Reinforcement Learning</b>		8 ECTS/LP
Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden sind mit Methoden und Prinzipien des Reinforcement Learning vertraut. Sie sind in der Lage, in kleinen Teams größere Projektaufgaben (Entwicklung von Softwaremodulen) zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.</p> <p><b>Schlüsselqualifikationen:</b> Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten; Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen; Verstehen von Teamprozessen; Fertigkeit der Zusammenarbeit in Teams; Fähigkeit zur Leitung von Teams; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Fähigkeit, Beiträge zur Wissenschaft zu leisten; Kompetenz zum Erkennen von bedeutend-technischen Entwicklungen; Qualitätsbewusstsein, Akribie.</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 240 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 90 Std. Praktikum (Präsenzstudium)</p>		
<p><b>Voraussetzungen:</b> Programmiererfahrung Modul Reinforcement Learning (INF-0207) - empfohlen</p>		
<p><b>Angebotshäufigkeit:</b> jedes Sommersemester</p>	<p><b>Empfohlenes Fachsemester:</b> ab dem 2.</p>	<p><b>Minimale Dauer des Moduls:</b> 1 Semester</p>
<p><b>SWS:</b> 6</p>	<p><b>Wiederholbarkeit:</b> siehe PO des Studiengangs</p>	
<p><b>Modulteile</b></p>		
<p><b>Modulteil: Praktikum Reinforcement Learning</b> <b>Lehrformen:</b> Praktikum <b>Sprache:</b> Deutsch <b>SWS:</b> 6</p>		
<p><b>Inhalte:</b> Die konkrete Aufgabenstellung aus dem Gebiet "Reinforcement Learning" wird jedes Semester neu entworfen.</p>		
<p><b>Zugeordnete Lehrveranstaltungen:</b></p> <p><b>Praktikum Reinforcement Learning (Praktikum)</b> Ziel der Veranstaltung ist, ausgewählte Inhalte und Algorithmen der Vorlesung "Reinforcement Learning" praktisch anzuwenden und umzusetzen. Die Teilnahme an dieser Veranstaltung ist deshalb von Vorteil, optional auch Kenntnisse aus dem Bereich Regelungstechnik. Im Verlauf des Praktikums arbeiten die Teilnehmer gemeinsam an Projekten im Kontext von HRI (Human-Robot Interaction) und HCI (Human-Computer Interaction). Das genaue Thema wird zu Beginn der Veranstaltung festgelegt. Die Teilnehmer des Praktikums werden in kleinere Gruppen aufgeteilt, die an unterschiedlichen Aufgabenbereichen arbeiten. Die Teilnehmerzahl ist aufgrund der Raumgröße und Hardwareverfügbarkeit auf max. 20 Studierende beschränkt.</p>		
<p><b>Prüfung</b> <b>Vortrag mit Softwarepräsentation, Ausarbeitung mit Softwaredokumentation</b> Projektarbeit</p>		

<b>Modul INF-0251: Seminar Artificial Intelligence</b>		4 ECTS/LP
Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: PD Dr. Jonghwa Kim		
<b>Inhalte:</b> Das Seminar wird als Blockseminar Ende Juni für SS oder Mitte Dezember für WS stattfinden. Der Themenbereich für dieses Seminar wird jährlich unter Berücksichtigung neuer Trends in "Künstliche Intelligenz und Intelligente Systeme" neu festgelegt.		
<b>Lernziele/Kompetenzen:</b> Nach dem Besuch des Seminars sind die Studierenden in der Lage, grundlegende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "Künstliche Intelligenz und Intelligente Systeme" selbstständig zu erarbeiten und bezogen auf ein spezielles Thema aus dem genannten Gebiet zu bewerten.  Sie verfügen über die Arbeitstechniken, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.  <b>Schlüsselqualifikationen:</b> Erlernen von Präsentationstechniken, Literaturrecherche, Arbeit mit englischer Fachliteratur, Grundsätze guter wissenschaftlicher Praxis		
<b>Arbeitsaufwand:</b> Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		<b>ECTS/LP-Bedingungen:</b> Bestehen der Modulprüfung
<b>Angebotshäufigkeit:</b> in der Regel mind. 1x pro Studienjahr	<b>Empfohlenes Fachsemester:</b> ab dem 1.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Modulteile</b>		
<b>Modulteil: Seminar Artificial Intelligence</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch / Englisch <b>SWS:</b> 2 <b>ECTS/LP:</b> 4.0		
<b>Inhalte:</b> Das Seminar wird als Blockseminar Ende Juni für SS oder Mitte Dezember für WS stattfinden. Der Themenbereich für dieses Seminar wird jährlich unter Berücksichtigung neuer Trends in "Künstliche Intelligenz und Intelligente Systeme" neu festgelegt.		
<b>Literatur:</b> aktuelle Forschungsliteratur		
<b>Zugeordnete Lehrveranstaltungen:</b> <b>Seminar Artificial Intelligence</b> (Seminar) Das Seminar Artificial Intelligence wird jedes Jahr als Blockseminar entweder Ende Juni zum SS oder Mitte Dezember zum WS stattfinden. Der Themenbereich für dieses Seminar wird jährlich unter Berücksichtigung neuer Trends in "Künstliche Intelligenz und Intelligente Systeme" neu festgelegt.		

---

**Prüfung**

**Vortrag und schriftliche Ausarbeitung**

Seminar

<b>Modul ZCS-2000: Softskills</b>		2 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Claudia Lange-Hetmann		
<p><b>Lernziele/Kompetenzen:</b></p> <p>Die Studierenden erwerben in diesem Modul - je nach Kurswahl - entweder kommunikative, soziale oder methodische Fähigkeiten, die unerlässlich für ihre künftige Berufsfähigkeit sind, denn diese fordert eine überzeugende Persönlichkeit des Einzelnen und eine einwandfreie und zielgerichtete Interaktion im Team. Daher wird bei der Auswahl empfohlen, einen Kurs aus einem der drei Kompetenzgebiete zu wählen, die zur Stärkung der eigenen Persönlichkeit sinnvoll und wichtig sind. Zudem bildet die interdisziplinäre Zusammensetzung der Teilnehmer aus unterschiedlichen Fachrichtungen den typischen Wirkungskreis im späteren Arbeitsumfeld ab.</p> <p>Die Studierenden werden - abhängig von der Kursthemenwahl -</p> <ul style="list-style-type: none"> <li>- neben dem Erwerb der Fertigkeit der verständlichen, sicheren und überzeugenden Darbietung von Ideen, Konzepten und Ergebnissen bzw. dem Verständnis der psychologischen Grundlagen von Dialogen und Verhandlungen dieses Wissen anwenden können, um Interesse, Verständlichkeit und Sympathie zu erzeugen und zielorientiert zu präsentieren bzw. zu argumentieren.</li> <li>- die Kommunikations-, Dialog- und Teamprozesse in Bezug auf Motivation, Effektivität und kennen die Entstehung, Dynamik, Lösung und Prävention von Konflikten verstehen und können Moderationstechniken und ihre Fertigkeit zur Selbstreflexion anwenden, sie beherrschen die Regeln bei der Teamarbeit, bei Besprechungen bis hin zur Führung von Teams oder kennen den Nutzen von gesellschaftlichem Engagement für sich und die Gesellschaft.</li> <li>- grundlegende Konzepte des Projektmanagements (u.a. Entwurf von strategischen Projektstrukturplänen, Analyse der Projektumwelt/-risiken, Projektcontrolling) verstehen und können die Grundlagen der Motivationspsychologie und zentrale Führungstechniken zur Erreichung des Projekterfolgs anwenden. Oder sie können grundlegende Strategien und Methoden für die Entwicklung und Absicherung einer Unternehmensführung anwenden, sie kennen Marketing- u. Vertriebsstrategien, bewerten deren Erfolgsaussichten und haben Kenntnisse in Personal- und Finanzmanagement. Sie verstehen Probleme zu analysieren und können konstruktiv im Team eine Lösung erarbeiten und kommunizieren.</li> </ul> <p>Besonderer Wert wird - je nach Kurs - auf die Weiterentwicklung der eigenen Präsentations- und Kommunikationsfähigkeit, der Teamkompetenz sowie die Anwendung des Methodenwissens und die Erreichung realistischer Ziele gelegt.</p> <p>Die interdisziplinäre Herangehensweise an eine Problemstellung wird durch die interdisziplinäre Zusammensetzung der Kleingruppen in den Kursen trainiert, durch praktische Übungen in den Kursen gefestigt und durch Selbstreflexion und Feedbackmethoden verinnerlicht.</p>		
<p><b>Bemerkung:</b></p> <p><u>Anmeldungspflicht:</u> Für die Teilnahme an den Kursen ist eine Anmeldung über digicampus/ VV erforderlich.</p> <p><u>Anmeldephase:</u> 2.Jan – 2.Feb (für das folgende SS) bzw. 1. Juli – 31. Juli (für das folgende WS).</p> <p>Die Kurse finden größtenteils ab Mitte März bis zum 2. Sa im April (SS) bzw. ab Mitte Sep. bis 2. Sa im Okt. statt und haben eine limitierte Teilnehmerzahl pro Semester.</p>		
<p><b>Arbeitsaufwand:</b></p> <p>Gesamt: 60 Std.</p> <p>20 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> <p>20 Std. Seminar (Präsenzstudium)</p> <p>5 Std. Vorbereitung von Präsentationen (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		<b>ECTS/LP-Bedingungen:</b> aktive Übungsteilnahme im Kurs
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b> ab dem 2.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	

<p><b>Moduleile</b></p>
<p><b>Moduleil: Softskills</b>  <b>Lehrformen:</b> Kurs  <b>Sprache:</b> Deutsch / Englisch  <b>SWS:</b> 2  <b>ECTS/LP:</b> 2.0</p>
<p><b>Inhalte:</b></p> <p>Zur Auswahl stehen nachfolgende Kurse/Themen:</p> <p><u>(1) Kommunikationskompetenz</u></p> <ul style="list-style-type: none"> <li>- Rhetorik</li> <li>- Präsentation</li> <li>- strategische Gesprächsführung</li> </ul> <p><u>(2) Sozialkompetenz</u></p> <ul style="list-style-type: none"> <li>- Konfliktmanagement</li> <li>- Moderation &amp; Teamleitung</li> <li>- Führungskompetenzen entwickeln</li> </ul> <p><u>(3) Methodenkompetenz</u></p> <ul style="list-style-type: none"> <li>- Zeit-/Selbst-/Changemanagement</li> <li>- Besprechungsmanagement</li> <li>- Innovationen entwickeln</li> <li>- Projektmanagement</li> <li>- Unternehmerisches Denken</li> </ul> <p>Weiterehin können auch Kompakt-Kurse gewählt werden, bei denen die Teilnehmer o.g. Fähigkeiten erlernen und eine Projektaufgabe im Team bearbeiten:</p> <p>KK Future Competencies          KK Märkte für Menschen          KK Projektbasiertes Unternehmertum          KK Entrepreneurship</p> <p>Der höhere Zeitaufwand wird mit mehr Erfahrung honoriert.</p> <p>Detailbeschreibungen zu allen Kursen sowie die konkreten Kursthemen und Termine pro Semester unter <a href="http://www.uni-augsburg.de/de/career-service/studierende/veranstaltungen_fakultaet/">http://www.uni-augsburg.de/de/career-service/studierende/veranstaltungen_fakultaet/</a> bzw. im VV Anmeldeesystem <a href="https://thi-vv.informatik.uni-augsburg.de/vv/">https://thi-vv.informatik.uni-augsburg.de/vv/</a></p>
<p><b>Literatur:</b></p> <p>(1) - Friedemann Schulz von Thun, miteinander reden 1-3, Rowohlt Taschenbuch -- Hütter,H., Degener,M.: Praxishandbuch PowerPoint-Präsentation, Gabler Verlag --- R. Fisher, W. Ury, B. Patton: Das Harvard-Konzept: Der Klassiker der Verhandlungstechnik, Campus Verlag, Frankfurt/New York</p> <p>(2) - Schwarz, G. (2001): Konfliktmanagement. Konflikte erkennen, analysieren, lösen. Wiesbaden. --- Hug, B.: Führen von Arbeitsgruppen. In: T. Steiger/ E. Lippmann (Hrsg.): Handbuch angewandte Psychologie für Führungskräfte. Berlin Heidelberg 1999, S.319-338 --- Andre Habisch, "Corporate Citizenship", Gesellschaftliches Engagement von Unternehmen in Deutschland</p> <p>(3) - Reinhold Westermann Georg Kraus: Projektmanagement mit System - Organisation, Methoden, Steuerung, Gabler Verlag 4. überarbeitete Auflage, 2010, ISBN-10:3-8349-1905-5 --- Bruno Jenny , Projektmanagement - Das Wissen für eine erfolgreiche Karriere, Vdf Hochschulverlag AG, Mai 2009 --- Fueglistaller, U.; Müller, C.; Müller, S.; Volery, T.: Entrepreneurship. Gabler Verlag 2012 --- Business Model Generation. Ein Handbuch für Visionäre, Spielveränderer und Herausforderer. Alexander Osterwalder &amp; Yves Pigneur, 2010. Campus Verlag.</p>
<p><b>Zugeordnete Lehrveranstaltungen:</b></p> <p><b>Softskill-Kurse/Kompakt-Kurse für Mathematik, Physik/MaWi, Wing - IngInf, Informatik (MPIng_I) (Kurs)</b></p>

---

**Prüfung**

**Anwesenheit und aktive Übungsteilnahme im Kurs**

Beteiligungsnachweis, unbenotet

<b>Modul INF-0003: Masterarbeit</b>		30 ECTS/LP
Version 1.0.0 (seit WS13/14) Modulverantwortliche/r: Alle Professorinnen und Professoren, die Veranstaltungen für diesen Studiengang anbieten.		
<p><b>Lernziele/Kompetenzen:</b> Die Studierenden sind mit der wissenschaftlichen Methodik sowie Techniken der Literaturrecherche vertraut, sind in der Lage, unter Anleitung praktische oder theoretische Methoden zur Bearbeitung eines vorgegebenen Themas einzusetzen und besitzen die Kompetenz, ein Problem der Informatik innerhalb einer vorgegebenen Frist weitgehend selbständig mit wissenschaftlichen Methoden zu bearbeiten sowie die Ergebnisse schriftlich und mündlich darzustellen.</p> <p>Darüber hinaus können die Studierenden grundlegende Strategien, Methoden und klare Vorstellungen entwickeln über ihre Ziele und Prioritäten, sie bewerten ihren persönlichen Arbeitsstil und schaffen eine effiziente Nutzung ihrer Ressourcen. Sie wenden Hilfsmittel und Techniken der Selbstorganisation an, die ihrem persönlichen Arbeitsstil entsprechen.</p> <p><b>Schlüsselqualifikationen:</b> Team- und Kommunikationsfähigkeit, Durchhaltevermögen, schriftliche und mündliche Darstellung eigener (praktischer oder theoretischer) Ergebnisse, Fertigkeit der effizienten Ressourcennutzung, Einschätzung der Relevanz eigener Ergebnisse, Grundsätze guter wissenschaftlicher Praxis, Fähigkeit zur Selbstreflexion und eines reflektierten Arbeitsstils, Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von eigenen und fremden (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und deren Dokumentation</p>		
<p><b>Arbeitsaufwand:</b> Gesamt: 900 Std. 15 Std. Seminar (Präsenzstudium) 885 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> nach Bedarf	<b>Empfohlenes Fachsemester:</b> 4.	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> siehe PO des Studiengangs	
<b>Moduleile</b>		
<b>Modulteil: Masterarbeit</b>		
<b>Sprache:</b> Deutsch		
<b>SWS:</b> 1		
<b>Inhalte:</b> entsprechend dem gewählten Thema		
<b>Literatur:</b> Die Festlegung der Literatur erfolgt abhängig vom konkreten Thema der Arbeit in Absprache mit dem Betreuer.		
<b>Prüfung</b>		
<b>Masterarbeit</b> Masterarbeit		



<b>Modul INF-0221: Anleitung zu wissenschaftlichen Arbeiten</b>		0 ECTS/LP
Version 1.0.0 Modulverantwortliche/r:		
<b>Lernziele/Kompetenzen:</b> Die Teilnehmer wissen, wie sie an wissenschaftliche Arbeiten heran gehen, welche Vorgehensweise sie ans Ziel führt und welche Maßstäbe gelten, damit ihre Arbeit als wissenschaftlich angesehen wird.		
<b>Bemerkung:</b> Dies ist eine freiwillige Veranstaltung und gibt <b>keine</b> ECTS-Punkte!		
<b>Arbeitsaufwand:</b> Gesamt: 15 Std.		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 1	<b>Wiederholbarkeit:</b> keine	
<b>Modulteile</b>		
<b>Modulteil: Anleitung zu wissenschaftlichen Arbeiten</b> <b>Sprache:</b> Deutsch <b>SWS:</b> 1		
<b>Inhalte:</b> Begleitung bei der Anfertigung von Seminar-/Bachelor-/Master-/Diplomarbeiten und Dissertationen.		

<b>Modul INF-0222: Oberseminar Informatik</b>		0 ECTS/LP
Version 1.0.0 Modulverantwortliche/r:		
<b>Lernziele/Kompetenzen:</b> Im Oberseminar werden wissenschaftliche Themen z.B. in Form von Abschlussarbeiten oder Vorträgen zu Praxis-/Forschungs-/Projektmodulen vorgestellt und diskutiert. Die Studierenden erhalten somit Einblicke in wissenschaftliches Arbeiten.		
<b>Bemerkung:</b> Dies ist eine freiwillige Veranstaltung und gibt <b>keine</b> ECTS-Punkte!		
<b>Arbeitsaufwand:</b> Gesamt: 30 Std. 30 Std. Seminar (Präsenzstudium)		
<b>Voraussetzungen:</b> keine		
<b>Angebotshäufigkeit:</b> jedes Semester	<b>Empfohlenes Fachsemester:</b>	<b>Minimale Dauer des Moduls:</b> 1 Semester
<b>SWS:</b> 2	<b>Wiederholbarkeit:</b> keine	
<b>Modulteile</b>		
<b>Modulteil: Oberseminar Informatik</b> <b>Lehrformen:</b> Seminar <b>Sprache:</b> Deutsch <b>SWS:</b> 2		