
Modulhandbuch

M.Sc. Informatik und Multimedia, PO 2018

Fakultät für Angewandte Informatik

Wintersemester 2018/2019

Studienbeginn ab Wintersemester 2018/2019

Liebe Studierenden,

willkommen im Master Informatik und Multimedia an der Universität Augsburg!

Die Prüfungsordnung für den M.Sc. Informatik wurde basierend auf den Erfahrungen der letzten Semester komplett überarbeitet. So ist der Studiengang übersichtlicher gegliedert und es passiert seltener, dass Leistungspunkte abgeschnitten werden (wenn ihr Module ungünstig wählt, kann das immernoch passieren, aber es ist nun leichter, sie passend zu wählen).

Die Informatik-Module sind nun den Modulgruppen „Theoretische Informatik“, „Praktische Informatik“, „Technische Informatik“ und „Multimedia“ zugeordnet. Aus den ersten drei Bereichen müsst ihr jeweils 5 LP erbringen. Wenn ihr pro Bereich ein Modul mit genau 5 LP wählt, habt ihr keinen Verschnitt. In der Modulgruppe „Multimedia“ besucht ihr Module mit insgesamt 26 LP, ohne Verschnitt z.B. mit 8+8+5+5 LP.

Alle Module aus diesen Bereichen findet ihr nochmal im „Wahlbereich Informatik und Multimedia“. Hier leistet ihr 35 LP mit Modulen, die ihr noch nicht in einem der anderen Bereiche belegt habt: ohne Verschnitt z.B. 8+8+6+6+5+2 LP = 35 LP.

Schließlich benötigt ihr noch genau ein Seminar (aus der gleichnamigen Modulgruppe), genau ein Projektmodul (aus der Modulgruppe „Projekt“) und die Masterarbeit, um euer Studium abzuschließen. Hier ist jeweils kein Verschnitt möglich.

In der vorläufigen Version des Modulhandbuchs war angekündigt, dass das Modul „INF-0148: Entwurf und Analyse fehlertolerierender Rechensysteme“ im WiSe 2018/2019 angeboten wird, jedoch muss dieses nun leider aufgrund mangelnder personeller Kapazitäten entfallen.

Da das Modulhandbuch ein Service für euch als Studierende ist, arbeiten wir eng mit der Studierendenvertretung Informatik zusammen. Solltet Ihr Anregungen, Fragen, Kritik oder Verbesserungsvorschläge zum neuen Modulhandbuch haben, so teilt diese einfach der Studierendenvertretung Informatik mit. Ihr erreicht sie unter fsinfo@informatik.uni-augsburg.de und persönlich im Raum 1007N.

Guten Start ins Master-Studium und

viele Grüße,

Euer Modulhandbuch-Beauftragter

Martin Frieb

ID	Modul	Semester	ECTS	SWS	Prüfung
Theoretische Informatik Module im Umfang von 5 LP sind zu erbringen.					
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-0056	Online-Algorithmen	unregelmäßig	5	2 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0116	Algebraische Semantik und Algebraische Systementwicklung	unregelmäßig	8	4 Vorlesung 2 Übung	Klausur 120Minuten
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-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

ID	Modul	Semester	ECTS	SWS	Prüfung
Praktische Informatik Module im Umfang von 5 LP sind zu erbringen.					
INF-0033	Modellgetriebene Softwareentwicklung	jedes Sommersemester	6	3 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0034	Softwarearchitekturen und Enterprise Architecture Management	jedes Wintersemester	6	3 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0037	Praktikum Automotive Software Engineering	jedes Semester	6	6 Praktikum	Mündliche Prüfung 30Minuten
INF-0038	Praktikum Avionic Software Engineering	jedes Semester	6	6 Praktikum	Mündliche Prüfung 30Minuten
INF-0066	Organic Computing II	jedes Sommersemester	5	2 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
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-0117	Funktionale Modellierung für Geoinformationssysteme	unregelmäßig	5	2 Vorlesung 2 Übung	Klausur 120Minuten
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	unregelmäßig (i. d. R. im WS)	8	2 Vorlesung 4 Übung	Mündliche Prüfung 45Minuten
INF-0133	Selbstorganisierende, adaptive Systeme	jedes Wintersemester	8	2 Vorlesung 4 Übung	Klausur 90Minuten
INF-0189	Qualitätssicherung im Software Engineering	unregelmäßig (i. d. R. im SoSe)	5	2 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0213	Parallele und Verteilte Datenbanksysteme	unregelmäßig	5	2 2	Mündliche Prüfung 30Minuten
INF-0217	Praktikum Autonomes Fahren	jedes Sommersemester	10	10 Praktikum	Mündliche Prüfung 30Minuten
INF-0233	Industrierobotik	jedes Semester	8	2 Vorlesung 4 Übung	praktische Prüfung 45Minuten
INF-0235	Software für Industrie 4.0	unregelmäßig (i. d. R. im SoSe)	5	2 Vorlesung 2 Übung	Mündliche Prüfung 45Minuten
INF-0248	Kollaborative Robotik	unregelmäßig	8	4 Übung 2 Vorlesung	praktische Prüfung 60Minuten
INF-0277	Analyzing Massive Data Sets	unregelmäßig (i. d. R. im SoSe)	8	4 Vorlesung 2 Übung	Klausur 90Minuten
INF-0284	Praktikum Mobile Application Development	jedes Sommersemester	5	4 Praktikum	Praktikum

INF-0291	Praktikum Selbstlernende Systeme	jedes Semester	8	2 Praktikum	Portfolioprfung (Kombination aus Praktischer und Schriftlich-Mündlicher Prfung)
ZCS-2000	Softskills	jedes Semester	2	2 Kurs	Beteiligungsnachweis

ID	Modul	Semester	ECTS	SWS	Prüfung
Technische Informatik Module im Umfang von 5 LP sind zu erbringen.					
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	Mündliche Prüfung 30Minuten
INF-0145	Mikrorechner-technik und Echtzeitsysteme	jedes Wintersemester	6	3 Vorlesung 1 Übung	Klausur 90Minuten
INF-0147	Prozessorarchitektur	unregelmäßig (i. d. R. im WS)	5	2 Vorlesung 2 Übung	Klausur 60Minuten
INF-0148	Entwurf und Analyse fehlertolerierender Rechensysteme	unregelmäßig	5	2 Vorlesung 2 Übung	Klausur 60Minuten
INF-0149	Praktikum Eingebettete Systeme	jedes Sommersemester	5	4 Praktikum	Praktikum
INF-0150	Hardware-Entwurf	unregelmäßig	8	2 Vorlesung 4 Übung	Praktikum
INF-0216	Vertiefte Multicore-Programmierung	unregelmäßig	8	2 Vorlesung 4 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0297	Praktikum Prozessorbau	einmalig WS	5	4 Praktikum	praktische Prüfung

ID	Modul	Semester	ECTS	SWS	Prüfung
Multimedia Module im Umfang von 26 LP sind zu erbringen.					
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-0112	Graphikprogrammierung	unregelmäßig	8	4 Vorlesung 2 Übung	Klausur 120Minuten
INF-0175	Multimedia I: Usability Engineering	jedes Wintersemester	8	4 Vorlesung 2 Übung	Übung + Praktikum
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-0182	Praktikum Multimodale Echtzeitsignalverarbeitung	jedes Sommersemester	8	6 Praktikum	Projektarbeit
INF-0183	Praktikum Spieleprogrammierung	jedes Wintersemester	8	6 Praktikum	Projektarbeit
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-0272	Intelligente Signalanalyse in der Medizin	jedes Sommersemester	5	2 Vorlesung 2 Übung	Klausur 90Minuten
INF-0273	Praktikum Mobile Sensing for Fitness and Wellbeing	jedes Wintersemester	5	4 Praktikum	Praktikum
INF-0279	Music Informatics	jedes Sommersemester	5	2 Vorlesung 2 Übung	Klausur 90Minuten
INF-0284	Praktikum Mobile Application Development	jedes Sommersemester	5	4 Praktikum	Praktikum
INF-0293	Advanced Deep Learning	jedes Semester	8	2 Vorlesung 4 Übung	Portfolioprfung (Die Endnote setzt sich aus bewerteten Übungsblättern und einem bewerteten Teamprojekt zusammen.)
INF-0294	Speech Pathology	jedes Wintersemester	5	2 Vorlesung 2 Übung	Klausur 90Minuten
INF-0296	Praktikum Interactive Machine Learning	jedes Semester	8	6 Praktikum	Projektarbeit

ID	Modul	Semester	ECTS	SWS	Prüfung
Wahlbereich Informatik und Multimedia Module im Umfang von 35 LP sind zu erbringen.					
INF-0033	Modellgetriebene Softwareentwicklung	jedes Sommersemester	6	3 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0034	Softwarearchitekturen und Enterprise Architecture Management	jedes Wintersemester	6	3 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0037	Praktikum Automotive Software Engineering	jedes Semester	6	6 Praktikum	Mündliche Prüfung 30Minuten
INF-0038	Praktikum Avionic Software Engineering	jedes Semester	6	6 Praktikum	Mündliche Prüfung 30Minuten
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-0056	Online-Algorithmen	unregelmäßig	5	2 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0066	Organic Computing II	jedes Sommersemester	5	2 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten

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	Mündliche Prüfung 30Minuten
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-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-0112	Graphikprogrammierung	unregelmäßig	8	4 Vorlesung 2 Übung	Klausur 120Minuten
INF-0116	Algebraische Semantik und Algebraische Systementwicklung	unregelmäßig	8	4 Vorlesung 2 Übung	Klausur 120Minuten
INF-0117	Funktionale Modellierung für Geoinformationssysteme	unregelmäßig	5	2 Vorlesung 2 Übung	Klausur 120Minuten
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	unregelmäßig (i. d. R. im WS)	8	2 Vorlesung 4 Übung	Mündliche Prüfung 45Minuten
INF-0133	Selbstorganisierende, adaptive Systeme	jedes Wintersemester	8	2 Vorlesung 4 Übung	Mündliche Prüfung 45Minuten
INF-0145	Mikrorechnertechnik und Echtzeitsysteme	jedes Wintersemester	6	3 Vorlesung 1 Übung	Klausur 90Minuten
INF-0147	Prozessorarchitektur	unregelmäßig (i. d. R. im WS)	5	2 Vorlesung 2 Übung	Klausur 60Minuten
INF-0148	Entwurf und Analyse fehlertolerierender Rechensysteme	unregelmäßig	5	2 Vorlesung 2 Übung	Klausur 60Minuten
INF-0149	Praktikum Eingebettete Systeme	jedes Sommersemester	5	4 Praktikum	Praktikum
INF-0150	Hardware-Entwurf	unregelmäßig	8	2 Vorlesung 4 Übung	Praktikum
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-0175	Multimedia I: Usability Engineering	jedes Wintersemester	8	4 Vorlesung 2 Übung	Übung + Praktikum
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-0182	Praktikum Multimodale Echtzeitsignalverarbeitung	jedes Sommersemester	8	6 Praktikum	Projektarbeit
INF-0183	Praktikum Spieleprogrammierung	jedes Wintersemester	8	6 Praktikum	Projektarbeit
INF-0189	Qualitätssicherung im Software Engineering	unregelmäßig (i. d. R. im SoSe)	5	2 Vorlesung 2 Übung	Mündliche Prüfung 30Minuten
INF-0198	Intelligente Systeme	unregelmäßig	5	2 Vorlesung 2 Übung	Klausur 90Minuten
INF-0201	Platzeffiziente Algorithmen	unregelmäßig	5	2 Vorlesung 2 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0207	Reinforcement Learning	jedes Wintersemester	8	2 Vorlesung 4 Übung	Projektarbeit (Projektarbeit / mündliche Prüfung)

					30Minuten
INF-0213	Parallele und Verteilte Datenbanksysteme	unregelmäßig	5	2 2	Mündliche Prüfung 30Minuten
INF-0216	Vertiefte Multicore-Programmierung	unregelmäßig	8	2 Vorlesung 4 Übung	Mündliche Prüfung (Dauer: 30-45 Minuten)
INF-0217	Praktikum Autonomes Fahren	jedes Sommersemester	10	10 Praktikum	Mündliche Prüfung 30Minuten
INF-0233	Industrierobotik	jedes Semester	8	2 Vorlesung 4 Übung	praktische Prüfung 45Minuten
INF-0235	Software für Industrie 4.0	unregelmäßig (i. d. R. im SoSe)	5	2 Vorlesung 2 Übung	Mündliche Prüfung 45Minuten
INF-0243	Process Mining	jedes Sommersemester	6	3 Vorlesung 1 Übung	Klausur 90Minuten
INF-0248	Kollaborative Robotik	unregelmäßig	8	4 Übung 2 Vorlesung	praktische Prüfung 60Minuten
INF-0250	Praktikum Reinforcement Learning	jedes Sommersemester	8	6 Praktikum	Projektarbeit Stunden
INF-0272	Intelligente Signalanalyse in der Medizin	jedes Sommersemester	5	2 Vorlesung 2 Übung	Klausur 90Minuten
INF-0273	Praktikum Mobile Sensing for Fitness and Wellbeing	jedes Wintersemester	5	4 Praktikum	Praktikum

INF-0277	Analyzing Massive Data Sets	unregelmäßig (i. d. R. im SoSe)	8	4 Vorlesung 2 Übung	Klausur 90Minuten
INF-0279	Music Informatics	jedes Sommersemester	5	2 Vorlesung 2 Übung	Klausur 90Minuten
INF-0284	Praktikum Mobile Application Development	jedes Sommersemester	5	4 Praktikum	Praktikum
INF-0291	Praktikum Selbstlernende Systeme	jedes Semester	8	2 Praktikum	Portfolioprüfung (Kombination aus Praktischer und Schriftlich- Mündlicher Prüfung)
INF-0293	Advanced Deep Learning	jedes Semester	8	2 Vorlesung 4 Übung	Portfolioprüfung (Die Endnote setzt sich aus bewerteten Übungsblättern und einem bewerteten Teamprojekt zusammen.)
INF-0294	Speech Pathology	jedes Wintersemester	5	2 Vorlesung 2 Übung	Klausur 90Minuten
INF-0296	Praktikum Interactive Machine Learning	jedes Semester	8	6 Praktikum	Projektarbeit
INF-0297	Praktikum Prozessorbau	einmalig WS	5	4 Praktikum	praktische Prüfung
ZCS-2000	Softskills	jedes Semester	2	2 Kurs	Beteiligungsnachweis

ID	Modul	Semester	ECTS	SWS	Prüfung
Seminar Module im Umfang von 4 LP sind zu erbringen.					
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-0058	Seminar Algorithmen und Datenstrukturen für Master	unregelmäßig	4	2 Seminar	Seminar
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-0084	Seminar Next Generation Networks	jedes Sommersemester	4	2 Seminar	Seminar
INF-0095	Seminar Multimedia Computing (MA)	jedes Sommersemester	4	2 Seminar	Seminar
INF-0118	Seminar Programmiermethodik und Multimediale Informationssysteme für Master	in der Regel mind. 1x pro Studienjahr	4	2 Seminar	Seminar
INF-0136	Seminar Software- und Systems Engineering (Master)	jedes Semester	4	2 Seminar	Seminar 45Minuten

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-0164	Seminar Theorie verteilter Systeme A	unregelmäßig	4	2 Seminar	Seminar
INF-0227	Seminar Datenbanksysteme für Master	unregelmäßig (i. d. R. im SoSe)	4	2 Seminar	Seminar
INF-0232	Seminar Medical Information Sciences (MA)	jedes Semester	4	2 Seminar	Seminar
INF-0240	Seminar Informationssysteme für Master	unregelmäßig (i. d. R. im WS)	4	2 Seminar	Seminar
INF-0251	Seminar Artificial Intelligence	in der Regel mind. 1x pro Studienjahr	4	2 Seminar	Seminar
INF-0274	Seminar Embedded Intelligence for Health Care and Wellbeing (Master)	jedes Semester	4	2 Seminar	Seminar
INF-0281	Seminar Sports Informatics (Master)	jedes Semester	4	2 Seminar	Seminar
INF-0283	Seminar Computer Audition (Master)	jedes Semester	4	2 Seminar	Seminar

ID	Modul	Semester	ECTS	SWS	Prüfung
	Projekt Module im Umfang von 10 LP sind zu erbringen.				
INF-0042	Projektmodul Softwaremethodiken für verteilte Systeme	jedes Semester	10	1 Praktikum	Praktikum
INF-0059	Projektmodul Theoretische Informatik	jedes Semester	10	1 Praktikum	Praktikum
INF-0072	Projektmodul Organic Computing	jedes Semester	10	1 Praktikum	Praktikum
INF-0080	Projektmodul Datenbanken und Informationssysteme	jedes Semester	10	1 Praktikum	Praktikum
INF-0085	Projektmodul Kommunikationssysteme	jedes Semester	10	1 Praktikum	Praktikum
INF-0096	Projektmodul Multimedia Computing	jedes Semester	10	1 Praktikum	Praktikum
INF-0108	Projektmodul Lehrprofessur für Informatik	jedes Semester	10	1 Praktikum	Praktikum
INF-0119	Projektmodul Programmiermethodik und Multimediale Informationssysteme	jedes Semester	10	1 Praktikum	Praktikum
INF-0137	Projektmodul Software- und Systems Engineering	jedes Semester	10	1 Praktikum	praktische Prüfung
INF-0154	Projektmodul Systemnahe Informatik und Kommunikationssysteme	jedes Semester	10	1 Praktikum	Praktikum

INF-0165	Projektmodul Theorie verteilter Systeme	jedes Semester	10	1	Projektarbeit
INF-0170	Projektmodul Human-Centered Multimedia	jedes Semester	10	1 Praktikum	Praktikum
INF-0275	Projektmodul Embedded Intelligence for Health Care and Wellbeing	jedes Semester	10	1 Praktikum	Praktikum

ID	Modul	Semester	ECTS	SWS	Prüfung
	Abschlussleistung Module im Umfang von 30 LP sind zu erbringen.				
INF-0003	Masterarbeit	jedes Semester	30	0	Masterarbeit

ID	Modul	Semester	ECTS	SWS	Prüfung
	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	jedes Semester	0	1	
INF-0222	Oberseminar Informatik	jedes Semester	0	2 Seminar	

Übersicht nach Modulgruppen

1) Theoretische Informatik ECTS: 5

Module im Umfang von 5 LP sind zu erbringen.

INF-0051: Algorithmen für NP-harte Probleme (8 ECTS/LP, Wahlpflicht).....	9
INF-0052: Einführung in die Komplexitätstheorie (5 ECTS/LP, Wahlpflicht).....	11
INF-0053: I/O-effiziente Algorithmen (5 ECTS/LP, Wahlpflicht).....	13
INF-0054: Datenstrukturen (8 ECTS/LP, Wahlpflicht) *	15
INF-0056: Online-Algorithmen (5 ECTS/LP, Wahlpflicht).....	17
INF-0116: Algebraische Semantik und Algebraische Systementwicklung (8 ECTS/LP, Wahlpflicht).....	19
INF-0130: Formale Methoden im Software Engineering (8 ECTS/LP, Wahlpflicht).....	20
INF-0156: Algebraische Beschreibung paralleler Prozesse (6 ECTS/LP, Wahlpflicht).....	22
INF-0157: Endliche Automaten (5 ECTS/LP, Wahlpflicht).....	24
INF-0161: Petrinetze - eine Theorie paralleler Systeme (5 ECTS/LP, Wahlpflicht) *	25
INF-0163: Verteilte Algorithmen (8 ECTS/LP, Wahlpflicht).....	27
INF-0201: Platzeffiziente Algorithmen (5 ECTS/LP, Wahlpflicht).....	29
INF-0243: Process Mining (6 ECTS/LP, Wahlpflicht).....	31

2) Praktische Informatik ECTS: 5

Module im Umfang von 5 LP sind zu erbringen.

INF-0033: Modellgetriebene Softwareentwicklung (6 ECTS/LP, Wahlpflicht).....	33
INF-0034: Softwarearchitekturen und Enterprise Architecture Management (6 ECTS/LP, Wahlpflicht) *	35
INF-0037: Praktikum Automotive Software Engineering (6 ECTS/LP, Wahlpflicht) *	37
INF-0038: Praktikum Avionic Software Engineering (6 ECTS/LP, Wahlpflicht) *	39
INF-0066: Organic Computing II (5 ECTS/LP, Wahlpflicht).....	42
INF-0077: Suchmaschinen (8 ECTS/LP, Wahlpflicht).....	44
INF-0078: Datenbankprogrammierung (Oracle) (5 ECTS/LP, Wahlpflicht) *	46
INF-0117: Funktionale Modellierung für Geoinformationssysteme (5 ECTS/LP, Wahlpflicht).....	48
INF-0129: Softwaretechnik II (8 ECTS/LP, Wahlpflicht) *	50
INF-0130: Formale Methoden im Software Engineering (8 ECTS/LP, Wahlpflicht).....	53

* = Im aktuellen Semester wird mindestens eine Lehrveranstaltung für dieses Modul angeboten

INF-0131: Software- und Systemsicherheit (8 ECTS/LP, Wahlpflicht).....	55
INF-0133: Selbstorganisierende, adaptive Systeme (8 ECTS/LP, Wahlpflicht) *	57
INF-0189: Qualitätssicherung im Software Engineering (5 ECTS/LP, Wahlpflicht).....	59
INF-0213: Parallele und Verteilte Datenbanksysteme (5 ECTS/LP, Wahlpflicht).....	61
INF-0217: Praktikum Autonomes Fahren (10 ECTS/LP, Wahlpflicht).....	63
INF-0233: Industrierobotik (8 ECTS/LP, Wahlpflicht) *	65
INF-0235: Software für Industrie 4.0 (5 ECTS/LP, Wahlpflicht).....	67
INF-0248: Kollaborative Robotik (8 ECTS/LP, Wahlpflicht) *	69
INF-0277: Analyzing Massive Data Sets (8 ECTS/LP, Wahlpflicht).....	72
INF-0284: Praktikum Mobile Application Development (5 ECTS/LP, Wahlpflicht).....	74
INF-0291: Praktikum Selbstlernende Systeme (8 ECTS/LP, Wahlpflicht) *	76
ZCS-2000: Softskills (2 ECTS/LP, Wahlpflicht) *	78

3) Technische Informatik ECTS: 5

Module im Umfang von 5 LP sind zu erbringen.

INF-0067: Peer-to-Peer und Cloud Computing (5 ECTS/LP, Wahlpflicht) *	81
INF-0069: Weiterführende Betriebssystemkonzepte (8 ECTS/LP, Wahlpflicht).....	83
INF-0145: Mikrorechnertechnik und Echtzeitsysteme (6 ECTS/LP, Wahlpflicht) *	85
INF-0147: Prozessorarchitektur (5 ECTS/LP, Wahlpflicht) *	87
INF-0148: Entwurf und Analyse fehlertolerierender Rechensysteme (5 ECTS/LP, Wahlpflicht).....	89
INF-0149: Praktikum Eingebettete Systeme (5 ECTS/LP, Wahlpflicht).....	91
INF-0150: Hardware-Entwurf (8 ECTS/LP, Wahlpflicht).....	93
INF-0216: Vertiefte Multicore-Programmierung (8 ECTS/LP, Wahlpflicht).....	95
INF-0297: Praktikum Prozessorbau (5 ECTS/LP, Wahlpflicht) *	97

4) Multimedia ECTS: 26

Module im Umfang von 26 LP sind zu erbringen.

INF-0088: Bayesian Networks (5 ECTS/LP, Wahlpflicht).....	99
INF-0092: Multimedia II: Machine Learning and Computer Vision (8 ECTS/LP, Wahlpflicht).....	101
INF-0093: Probabilistic Robotics (5 ECTS/LP, Wahlpflicht) *	103
INF-0112: Graphikprogrammierung (8 ECTS/LP, Wahlpflicht).....	105
INF-0175: Multimedia I: Usability Engineering (8 ECTS/LP, Wahlpflicht) *	107
INF-0178: Praktikum Usability Engineering (8 ECTS/LP, Wahlpflicht).....	109

INF-0179: Einführung in die Spieleprogrammierung (8 ECTS/LP, Wahlpflicht).....	110
INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung (8 ECTS/LP, Wahlpflicht).....	112
INF-0183: Praktikum Spieleprogrammierung (8 ECTS/LP, Wahlpflicht) *	113
INF-0198: Intelligente Systeme (5 ECTS/LP, Wahlpflicht).....	115
INF-0207: Reinforcement Learning (8 ECTS/LP, Wahlpflicht) *	116
INF-0250: Praktikum Reinforcement Learning (8 ECTS/LP, Wahlpflicht).....	118
INF-0272: Intelligente Signalanalyse in der Medizin (5 ECTS/LP, Wahlpflicht).....	119
INF-0273: Praktikum Mobile Sensing for Fitness and Wellbeing (5 ECTS/LP, Wahlpflicht) *	121
INF-0279: Music Informatics (5 ECTS/LP, Wahlpflicht).....	123
INF-0284: Praktikum Mobile Application Development (5 ECTS/LP, Wahlpflicht).....	125
INF-0293: Advanced Deep Learning (8 ECTS/LP, Wahlpflicht) *	127
INF-0294: Speech Pathology (5 ECTS/LP, Wahlpflicht) *	129
INF-0296: Praktikum Interactive Machine Learning (8 ECTS/LP, Wahlpflicht) *	131

5) Wahlbereich Informatik und Multimedia ECTS: 35

Module im Umfang von 35 LP sind zu erbringen.

INF-0033: Modellgetriebene Softwareentwicklung (6 ECTS/LP, Wahlpflicht).....	133
INF-0034: Softwarearchitekturen und Enterprise Architecture Management (6 ECTS/LP, Wahlpflicht) *	135
INF-0037: Praktikum Automotive Software Engineering (6 ECTS/LP, Wahlpflicht) *	137
INF-0038: Praktikum Avionic Software Engineering (6 ECTS/LP, Wahlpflicht) *	139
INF-0051: Algorithmen für NP-harte Probleme (8 ECTS/LP, Wahlpflicht).....	142
INF-0052: Einführung in die Komplexitätstheorie (5 ECTS/LP, Wahlpflicht).....	144
INF-0053: I/O-effiziente Algorithmen (5 ECTS/LP, Wahlpflicht).....	146
INF-0054: Datenstrukturen (8 ECTS/LP, Wahlpflicht) *	148
INF-0056: Online-Algorithmen (5 ECTS/LP, Wahlpflicht).....	150
INF-0066: Organic Computing II (5 ECTS/LP, Wahlpflicht).....	152
INF-0067: Peer-to-Peer und Cloud Computing (5 ECTS/LP, Wahlpflicht) *	154
INF-0069: Weiterführende Betriebssystemkonzepte (8 ECTS/LP, Wahlpflicht).....	156
INF-0077: Suchmaschinen (8 ECTS/LP, Wahlpflicht).....	158
INF-0078: Datenbankprogrammierung (Oracle) (5 ECTS/LP, Wahlpflicht) *	160
INF-0088: Bayesian Networks (5 ECTS/LP, Wahlpflicht).....	162

INF-0092: Multimedia II: Machine Learning and Computer Vision (8 ECTS/LP, Wahlpflicht).....	164
INF-0093: Probabilistic Robotics (5 ECTS/LP, Wahlpflicht) *	166
INF-0112: Graphikprogrammierung (8 ECTS/LP, Wahlpflicht).....	168
INF-0116: Algebraische Semantik und Algebraische Systementwicklung (8 ECTS/LP, Wahlpflicht).....	170
INF-0117: Funktionale Modellierung für Geoinformationssysteme (5 ECTS/LP, Wahlpflicht).....	171
INF-0129: Softwaretechnik II (8 ECTS/LP, Wahlpflicht) *	173
INF-0130: Formale Methoden im Software Engineering (8 ECTS/LP, Wahlpflicht).....	176
INF-0131: Software- und Systemsicherheit (8 ECTS/LP, Wahlpflicht).....	178
INF-0133: Selbstorganisierende, adaptive Systeme (8 ECTS/LP, Wahlpflicht) *	180
INF-0145: Mikrorechnerarchitektur und Echtzeitsysteme (6 ECTS/LP, Wahlpflicht) *	182
INF-0147: Prozessorarchitektur (5 ECTS/LP, Wahlpflicht) *	184
INF-0148: Entwurf und Analyse fehlertolerierender Rechensysteme (5 ECTS/LP, Wahlpflicht).....	186
INF-0149: Praktikum Eingebettete Systeme (5 ECTS/LP, Wahlpflicht).....	188
INF-0150: Hardware-Entwurf (8 ECTS/LP, Wahlpflicht).....	190
INF-0156: Algebraische Beschreibung paralleler Prozesse (6 ECTS/LP, Wahlpflicht).....	192
INF-0157: Endliche Automaten (5 ECTS/LP, Wahlpflicht).....	194
INF-0161: Petrinetze - eine Theorie paralleler Systeme (5 ECTS/LP, Wahlpflicht) *	195
INF-0163: Verteilte Algorithmen (8 ECTS/LP, Wahlpflicht).....	197
INF-0175: Multimedia I: Usability Engineering (8 ECTS/LP, Wahlpflicht) *	199
INF-0178: Praktikum Usability Engineering (8 ECTS/LP, Wahlpflicht).....	201
INF-0179: Einführung in die Spieleprogrammierung (8 ECTS/LP, Wahlpflicht).....	202
INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung (8 ECTS/LP, Wahlpflicht).....	204
INF-0183: Praktikum Spieleprogrammierung (8 ECTS/LP, Wahlpflicht) *	205
INF-0189: Qualitätssicherung im Software Engineering (5 ECTS/LP, Wahlpflicht).....	207
INF-0198: Intelligente Systeme (5 ECTS/LP, Wahlpflicht).....	209
INF-0201: Platzeffiziente Algorithmen (5 ECTS/LP, Wahlpflicht).....	210
INF-0207: Reinforcement Learning (8 ECTS/LP, Wahlpflicht) *	212
INF-0213: Parallele und Verteilte Datenbanksysteme (5 ECTS/LP, Wahlpflicht).....	214
INF-0216: Vertiefte Multicore-Programmierung (8 ECTS/LP, Wahlpflicht).....	216
INF-0217: Praktikum Autonomes Fahren (10 ECTS/LP, Wahlpflicht).....	218

INF-0233: Industrierobotik (8 ECTS/LP, Wahlpflicht) *	220
INF-0235: Software für Industrie 4.0 (5 ECTS/LP, Wahlpflicht)	222
INF-0243: Process Mining (6 ECTS/LP, Wahlpflicht)	224
INF-0248: Kollaborative Robotik (8 ECTS/LP, Wahlpflicht) *	226
INF-0250: Praktikum Reinforcement Learning (8 ECTS/LP, Wahlpflicht)	229
INF-0272: Intelligente Signalanalyse in der Medizin (5 ECTS/LP, Wahlpflicht)	230
INF-0273: Praktikum Mobile Sensing for Fitness and Wellbeing (5 ECTS/LP, Wahlpflicht) *	232
INF-0277: Analyzing Massive Data Sets (8 ECTS/LP, Wahlpflicht)	234
INF-0279: Music Informatics (5 ECTS/LP, Wahlpflicht)	236
INF-0284: Praktikum Mobile Application Development (5 ECTS/LP, Wahlpflicht)	238
INF-0291: Praktikum Selbstlernende Systeme (8 ECTS/LP, Wahlpflicht) *	240
INF-0293: Advanced Deep Learning (8 ECTS/LP, Wahlpflicht) *	242
INF-0294: Speech Pathology (5 ECTS/LP, Wahlpflicht) *	244
INF-0296: Praktikum Interactive Machine Learning (8 ECTS/LP, Wahlpflicht) *	246
INF-0297: Praktikum Prozessorbau (5 ECTS/LP, Wahlpflicht) *	248
ZCS-2000: Softskills (2 ECTS/LP, Wahlpflicht) *	250

6) Seminar ECTS: 4

Module im Umfang von 4 LP sind zu erbringen.

INF-0039: Seminar über Software Engineering verteilter Systeme (MA) (4 ECTS/LP, Wahlpflicht) *	253
INF-0040: Seminar Grundlagen des Software Engineering für Automotive Systems (MA) (4 ECTS/LP, Wahlpflicht) *	255
INF-0041: Seminar Grundlagen des Software Engineering für Avionic Systems (MA) (4 ECTS/LP, Wahlpflicht) *	257
INF-0058: Seminar Algorithmen und Datenstrukturen für Master (4 ECTS/LP, Wahlpflicht)	259
INF-0070: Seminar Organic Computing (4 ECTS/LP, Wahlpflicht) *	260
INF-0071: Seminar Naturalanaloge Algorithmen und Multiagentensysteme (4 ECTS/LP, Wahlpflicht)	262
INF-0084: Seminar Next Generation Networks (4 ECTS/LP, Wahlpflicht)	264
INF-0095: Seminar Multimedia Computing (MA) (4 ECTS/LP, Wahlpflicht)	266
INF-0118: Seminar Programmiermethodik und Multimediale Informationssysteme für Master (4 ECTS/LP, Wahlpflicht) *	267
INF-0136: Seminar Software- und Systems Engineering (Master) (4 ECTS/LP, Wahlpflicht) *	269

* = Im aktuellen Semester wird mindestens eine Lehrveranstaltung für dieses Modul angeboten

INF-0152: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen (4 ECTS/LP, Wahlpflicht).....	271
INF-0153: Seminar Safety-Critical Systems (4 ECTS/LP, Wahlpflicht) *	273
INF-0164: Seminar Theorie verteilter Systeme A (4 ECTS/LP, Wahlpflicht).....	275
INF-0227: Seminar Datenbanksysteme für Master (4 ECTS/LP, Wahlpflicht).....	277
INF-0232: Seminar Medical Information Sciences (MA) (4 ECTS/LP, Wahlpflicht) *	279
INF-0240: Seminar Informationssysteme für Master (4 ECTS/LP, Wahlpflicht) *	281
INF-0251: Seminar Artificial Intelligence (4 ECTS/LP, Wahlpflicht) *	283
INF-0274: Seminar Embedded Intelligence for Health Care and Wellbeing (Master) (4 ECTS/LP, Wahlpflicht) *	285
INF-0281: Seminar Sports Informatics (Master) (4 ECTS/LP, Wahlpflicht) *	287
INF-0283: Seminar Computer Audition (Master) (4 ECTS/LP, Wahlpflicht) *	289

7) Projekt ECTS: 10

Module im Umfang von 10 LP sind zu erbringen.

INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme (10 ECTS/LP, Wahlpflicht) *	291
INF-0059: Projektmodul Theoretische Informatik (10 ECTS/LP, Wahlpflicht) *	292
INF-0072: Projektmodul Organic Computing (10 ECTS/LP, Wahlpflicht) *	293
INF-0080: Projektmodul Datenbanken und Informationssysteme (10 ECTS/LP, Wahlpflicht) *	295
INF-0085: Projektmodul Kommunikationssysteme (10 ECTS/LP, Wahlpflicht) *	296
INF-0096: Projektmodul Multimedia Computing (10 ECTS/LP, Wahlpflicht) *	298
INF-0108: Projektmodul Lehrprofessur für Informatik (10 ECTS/LP, Wahlpflicht) *	300
INF-0119: Projektmodul Programmiermethodik und Multimediale Informationssysteme (10 ECTS/LP, Wahlpflicht) *	302
INF-0137: Projektmodul Software- und Systems Engineering (10 ECTS/LP, Wahlpflicht) *	303
INF-0154: Projektmodul Systemnahe Informatik und Kommunikationssysteme (10 ECTS/LP, Wahlpflicht) *	305
INF-0165: Projektmodul Theorie verteilter Systeme (10 ECTS/LP, Wahlpflicht) *	307
INF-0170: Projektmodul Human-Centered Multimedia (10 ECTS/LP, Wahlpflicht) *	308
INF-0275: Projektmodul Embedded Intelligence for Health Care and Wellbeing (10 ECTS/LP, Wahlpflicht) *	309

8) Abschlussleistung ECTS: 30

Module im Umfang von 30 LP sind zu erbringen.

INF-0003: Masterarbeit (30 ECTS/LP, Wahlpflicht).....	310
---	-----

9) 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) *	312
INF-0222: Oberseminar Informatik (0 ECTS/LP, Wahlfach) *	313

Modul INF-0051: Algorithmen für NP-harte Probleme		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: Kenntnis verschiedener algorithmischer Lösungsansätze für NP-harte Probleme und die Fähigkeit, diese sinnvoll im Kontext neuer Probleme einzusetzen, zu analysieren und zu bewerten.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.		
Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 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)		
Voraussetzungen: Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes, insbesondere im Bereich der Graphenalgorithmen. Modul Informatik 3 (INF-0111) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Algorithmen für NP-harte Probleme (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4		
Inhalte: 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.		
Literatur: <ul style="list-style-type: none"> • Skript • Ausiello et al., Complexity and Approximation, Springer, Berlin, 1999. • J. Hromkovic, Algorithmics for Hard Problems, Springer, Berlin, 2001. 		
Modulteil: Algorithmen für NP-harte Probleme (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 2		

Prüfung

Algorithmen für NP-harte Probleme (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0052: Einführung in die Komplexitätstheorie		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: Verständnis für zentrale Themen und Methoden der Komplexitätstheorie; die Fähigkeit, komplexitätstheoretische Sachverhalte und Fragen präzise zu formulieren und zu untersuchen.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.		
Arbeitsaufwand: 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)		
Voraussetzungen: 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		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Einführung in die Komplexitätstheorie (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2		
Inhalte: 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.		
Literatur: <ul style="list-style-type: none"> • Skript • Christos H. Papadimitriou, Computational Complexity, Addison-Wesley, Reading, Mass., 1994. 		
Modulteil: Einführung in die Komplexitätstheorie (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 2		

Prüfung

Einführung in die Komplexitätstheorie (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0053: I/O-effiziente Algorithmen		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<p>Lernziele/Kompetenzen: 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.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.</p>		
<p>Arbeitsaufwand: 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>Voraussetzungen: Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes. Modul Informatik 3 (INF-0111) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 4</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: I/O-effiziente Algorithmen (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: 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".</p>		
<p>Literatur:</p> <ul style="list-style-type: none"> • Skript • J.S. Vitter, Algorithms and data structures for external memory, Foundations and Trends in Theoretical Computer Science 2 (2008), pp. 305-474 		

Modulteil: I/O-effiziente Algorithmen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

I/O-effiziente Algorithmen (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Beschreibung:

Die Prüfung findet zum Ende des Semesters statt. Es besteht bei Bedarf die Möglichkeit einer Nachholprüfung im Folgesemester.

Modul INF-0054: Datenstrukturen		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: Kenntnis einer Reihe von nichtelementaren Datenstrukturen und ihrer Analyse; Fähigkeit zur Anpassung dieser Datenstrukturen an neue Anwendungen und zur Entwicklung neuer einfacher Datenstrukturen.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.		
Arbeitsaufwand: 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)		
Voraussetzungen: Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes. Modul Informatik 3 (INF-0111) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
Modulteil: Datenstrukturen (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4
Inhalte: 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.
Literatur: Skript
Zugeordnete Lehrveranstaltungen: Datenstrukturen (Vorlesung) 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.

Modulteil: Datenstrukturen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Datenstrukturen (Übung)

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.

Prüfung

Datenstrukturen (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0056: Online-Algorithmen		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: 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.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.		
Arbeitsaufwand: 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)		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Online-Algorithmen (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2		
Inhalte: 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.		
Literatur: <ul style="list-style-type: none"> • Skript • A. Borodin und R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998. 		
Modulteil: Online-Algorithmen (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 2		

Prüfung

Online-Algorithmen (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0116: Algebraische Semantik und Algebraische Systementwicklung		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p>Lernziele/Kompetenzen: 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>Schlüsselqualifikationen: analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis</p>		
<p>Arbeitsaufwand: 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 anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p>		
Voraussetzungen: Modul Diskrete Strukturen für Informatiker (INF-0109) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Algebraische Semantik und Algebraische Systementwicklung (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 4		
Inhalte: Halbringe, Testelemente, Modale Operatoren, Iterationsoperatoren, Terminierungsanalyse, Wissens-/ Glaubenslogiken, Temporale Logiken, Algebra paralleler Systeme.		
Literatur: <ul style="list-style-type: none"> • Eigenes Skriptum • U. Hebisch, H. J. Weinert: Halbringe - Algebraische Theorie und Anwendungen in der Informatik, Teubner 1993 		
Modulteil: Algebraische Semantik und Algebraische Systementwicklung (Übung)		
Lehrformen: Übung		
Sprache: Deutsch		
SWS: 2		
Prüfung		
Algebraische Semantik und Algebraische Systementwicklung (Klausur) Klausur / Prüfungsdauer: 120 Minuten		

Modul INF-0130: Formale Methoden im Software Engineering		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
Lernziele/Kompetenzen: Fachbezogene Kompetenzen: <p>Die Studierenden können mathematisch-formale Methoden für die Programmverifikation, speziell bei sicherheitskritischer Software einsetzen. Durch das Verständnis von Kalkülen trainieren sie die Fertigkeit zum logischen, konzeptuellen und analytischen Denken. Sie können Spezifikationen von Datenstrukturen entwerfen und deren Eigenschaften formal beweisen. Sie sind in der Lage, funktionale Eigenschaften von sequentiellen und nebenläufigen Programmen zu formulieren und dafür Beweise zu erstellen. Sie erwerben die Fertigkeit zur abstrakten, modularen Modellierung von Softwaresystemen und zur systematischen Entwicklung von korrekten Programmen aus diesen Modellen. Sie erwerben dadurch die Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen und können mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen abgeben. Sie erwerben Fähigkeiten, um Beiträge zur Wissenschaft zu leisten.</p> Schlüsselqualifikationen: <ul style="list-style-type: none"> • Training des logischen Denkens • Analytisch-methodische Kompetenz • Erwerb von Abstraktionsfähigkeiten • Fähigkeiten zur Modellbildung für Softwaresysteme 		
Arbeitsaufwand: 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)		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Formale Methoden im Software Engineering (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2		
Inhalte: <p>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</p>		

Literatur:

- Sperschneider, Antoniou: Logic: A Foundation for Computer Science, Addison Wesley 1991
- Loeckx, Ehrich, Wolf: Specification of Abstract Data Types, Wiley 1996
- Ausführliche Dokumentation
- Folienhandout

Modulteil: Formale Methoden im Software Engineering (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Formale Methoden im Software Engineering (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Modul INF-0156: Algebraische Beschreibung paralleler Prozesse		6 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p>Lernziele/Kompetenzen:</p> <p>Die Studierenden kennen die Prozessalgebra CCS und formale Beschreibungen verteilter Systeme in CCS. Sie kennen einen Mechanismus, mit dem man in CCS die operationale Semantik definiert, und können diesen Mechanismus auf konkrete Prozesse anwenden. Sie besitzen die Fähigkeit, verteilte Systeme entsprechend auf eine exakte, algebraische Weise zu modellieren, d.h. einen entsprechenden Prozess zu entwerfen.</p> <p>Sie wissen, welche Anforderungen man in diesem Kontext an Äquivalenzbegriffe stellen muss und lernen verschiedene solche Begriffe kennen. Sie können auf verschiedenen Weisen -- z.B. auch durch algebraische Gesetze -- formal prüfen, ob ein System eine, ebenfalls in CCS geschriebene, Spezifikation erfüllt.</p> <p>Sie können beurteilen, ob ein eine algebraische Gleichung ein gültiges Gesetz ist. Zur Vertiefung der Kenntnisse werden die meisten Resultate der Vorlesung bewiesen.</p> <p>Die Studierenden wissen, wie man CCS und die operationale Semantik in einer spezifischen Weise um Zeit erweitern kann, und sind dadurch in der Lage, auch andere Prozessalgebren mit anderen Semantik-Regeln zu verstehen und zu verwenden.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Methodik; Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit zur Analyse und Strukturierung von Informatikproblemstellungen; Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 180 Std.</p> <p>15 Std. Übung (Präsenzstudium)</p> <p>45 Std. Vorlesung (Präsenzstudium)</p> <p>75 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>22 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>23 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
<p>Voraussetzungen:</p> <p>Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen</p> <p>Modul Logik für Informatiker (INF-0155) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: ab dem 5.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 4</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Algebraische Beschreibung paralleler Prozesse (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch</p> <p>SWS: 3</p> <p>Inhalte:</p> <p>Algebraische Spezifikation verteilter Systeme mittels der Prozessalgebra CCS; operationale Semantik mittels SOS-Regeln; Äquivalenz- bzw. Kongruenzbegriffe (starke und schwache Bisimulation, Beobachtungskongruenz); Nachweis von Kongruenzen mittels Axiomen; Einführung in eine Kombination von Bisimulation und Effizienzvergleich.</p> <p>Die meisten Resultate der Vorlesung werden bewiesen.</p>		

Literatur:

- R. Milner: Communication and Concurrency, Prentice Hall
- L. Aceto, A. Ingólfssdóttir, K.G. Larsen, J. Srba: Reactive Systems. Cambridge University Press 2007
- J. Bergstra, A. Ponse, S. Smolka (eds.): Handbook of Process Algebras, Elsevier

Modulteil: Algebraische Beschreibung paralleler Prozesse (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 1

Prüfung

Algebraische Beschreibung paralleler Prozesse (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0157: Endliche Automaten		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
Lernziele/Kompetenzen: Nach der Teilnahme haben die Studierenden vertiefte Kenntnisse in der Automatentheorie und können sie verständlich erläutern. Sie können deterministische Automaten minimieren und das Verfahren mit guter Effizienz automatisieren. Sie können Schaltkreisverhalten und Mealy-Automaten ineinander übersetzen, und sie können mit geeigneten Ergebnissen reguläre von nicht-regulären Sprachen unterscheiden. Zur Vertiefung der Kenntnisse werden die meisten Resultate der Vorlesung bewiesen. Sie haben anspruchsvolle Modellierungen von Problemen mit endlichen Automaten kennengelernt und können sich in neue Anwendungen der Automatentheorie einarbeiten und ihre Korrektheit beurteilen.		
Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Methodik; Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Auswahl und sichere Anwendung geeigneter Methoden; Qualitätsbewusstsein, Akribie		
Arbeitsaufwand: Gesamt: 150 Std. 20 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 37 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 48 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 45 Std. Vorlesung (Präsenzstudium)		
Voraussetzungen: Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen Modul Informatik 3 (INF-0111) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 5.	Minimale Dauer des Moduls: 1 Semester
SWS: 3	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Endliche Automaten (Vorlesung mit integrierter Übung) Lehrformen: Vorlesung + Übung Sprache: Deutsch SWS: 3		
Inhalte: Die Vorlesung vertieft die Kenntnisse über Endliche Automaten aus der Grundvorlesung "Einführung in die theoretische Informatik", insbesondere über Minimierung und Abschlusseigenschaften. Sie behandelt eine Anwendung bei der Lösung diophantischer Gleichungen. Sie stellt Mealy-, Moore- und Büchi-Automaten vor und behandelt die Übersetzungen zwischen Mealy-Automaten und Schaltkreisverhalten. Die meisten Resultate der Vorlesung werden bewiesen.		
Literatur: <ul style="list-style-type: none"> • Hopcroft, (Motwani, Ullman: Introduction to Automata Theory, Languages and Computation; deutsch: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie • Schöning: Theoretische Informatik kurz gefasst. 5. Auflage • Thomas: Automata on Infinite Objects. Chapter 4 in Handbook of Theoretical Computer Science, Hrsg. van Leeuwen 		
Prüfung Endliche Automaten (mündliche Prüfung) Mündliche Prüfung / Prüfungsdauer: 30 Minuten		

Modul INF-0161: Petrinetze - eine Theorie paralleler Systeme		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p>Lernziele/Kompetenzen: Die Studierenden kennen verschiedene Varianten von Petrinetzen und entsprechende formale, Graphen-basierte Beschreibungen paralleler bzw. nebenläufiger Systeme. Anhand verschiedener Verhaltensbegriffe (Schalt- und Schrittfolgen, Sprache, Failure-Semantik) lernen sie die neuartigen Aspekte der Abläufe solcher Systeme kennen. Sie werden befähigt, wichtige Systemeigenschaften mit Petrinetz-spezifischen Methoden zu beweisen oder zu widerlegen. Auf dieser Grundlage können sie parallele bzw. nebenläufige Systeme mit Petrinetzen formal modellieren sowie analysieren, unter welchen Aspekten sich verschiedene Modelle gleichen oder unterscheiden. Sie lernen einige Entscheidbarkeitsprobleme kennen und können eigene Beweise dazu führen. Zur Vertiefung der Kenntnisse werden die meisten Resultate der Vorlesung bewiesen.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Methodik; Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fertigkeit zur Analyse und Strukturierung von Informatikproblemstellungen; Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen		
Angebotshäufigkeit: unregelmäßig (i. d. R. im WS)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Petrinetze - eine Theorie paralleler Systeme (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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); Entscheidbarkeitsprobleme. Die meisten Resultate der Vorlesung werden bewiesen.</p>		
<p>Literatur:</p> <ul style="list-style-type: none"> • Desel, Reisig, Rozenberg (eds.): Lectures on Concurrency and Petri Nets. Advances in Petri Nets. Springer, LNCS 3098 • Peterson: Petri Net Theory and the Modelling of Systems. Prentice Hall • Reisig: Petrinetze - Eine Einführung. 2. Auflage; Springer 		
Zugeordnete Lehrveranstaltungen:		
Petrinetze -- eine Theorie paralleler Systeme (Vorlesung)		

Petrinetze bilden ein formales Hilfsmittel zur Modellierung paralleler Systeme, das durch die graphische Darstellung der Netze besonders attraktiv ist. Die Vorlesung behandelt verschiedene wichtige Klassen von Petrinetzen und zugehörigen Analyseverfahren. Dabei stellt sich die Frage, wie das Verhalten von parallelen Systemen auf einer abstrakten Ebene geeignet zu beschreiben ist; im Rahmen der Theorie der Petrinetze sind hier besonders Beschreibungen wichtig, bei denen ein Ablauf eines Systems durch eine Halbordnung der ausgeführten Aktionen modelliert wird.

Modulteil: Petrinetze - eine Theorie paralleler Systeme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Petrinetze -- eine Theorie paralleler Systeme (Übung)

Prüfung

Petrinetze - eine Theorie paralleler Systeme (Mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0163: Verteilte Algorithmen		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p>Lernziele/Kompetenzen: Die Studierenden lernen formale Modelle zur Beschreibung verteilter Algorithmen in synchronen und asynchronen Netzwerke sowie die entsprechenden Methoden zum Nachweis der Korrektheit und des Aufwands kennen. Sie können beschreiben, wie Fehler in den beiden Modellen behandelt werden. Die Studierenden lernen wichtige Algorithmen und ihren Aufwand kennen, können sie in konkreten Netzwerken simulieren und verstehen ihre Korrektheit. Sie werden befähigt, solche Algorithmen zu modifizieren und zu analysieren, so dass sie zugehörige Korrektheitsbeweise und Aufwandsbestimmungen prüfen oder selbst entwickeln können. Sie erlangen damit ein vertieftes Verständnis für die Probleme und Problemlösungen in verteilten Systemen.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Methodik; 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</p>		
<p>Arbeitsaufwand: 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)</p>		
<p>Voraussetzungen: Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen Modul Informatik 3 (INF-0111) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	

Modulteile		
Modulteil: Verteilte Algorithmen (Vorlesung)		
<p>Lehrformen: Vorlesung Sprache: Deutsch SWS: 4</p>		
<p>Inhalte: 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 nachgewiesen sowie Korrektheitsbeweise geführt.</p>		
<p>Literatur: Nancy Lynch, Distributed Algorithms. Morgan Kaufmann 1996</p>		
Modulteil: Verteilte Algorithmen (Übung)		
<p>Lehrformen: Übung Sprache: Deutsch SWS: 2</p>		

Prüfung

Verteilte Algorithmen (Mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0201: Platzeffiziente Algorithmen		5 ECTS/LP
Version 1.0.0 (seit WS15/16) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<p>Lernziele/Kompetenzen: 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>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.</p>		
<p>Arbeitsaufwand: 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>		
<p>Voraussetzungen: Teilnehmer der Vorlesung sollten über gute algorithmische Kenntnisse verfügen, insbesondere im Bereich Graphenalgorithmen. Modul Informatik 3 (INF-0111) - empfohlen</p>		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Platzeffiziente Algorithmen (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: 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>Literatur:</p> <ul style="list-style-type: none"> • Skript 		

Modulteil: Platzeffiziente Algorithmen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Platzeffiziente Algorithmen (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0243: Process Mining <i>Process Mining</i>		6 ECTS/LP
Version 1.1.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Robert Lorenz		
<p>Lernziele/Kompetenzen: Teilnehmer verstehen die folgenden wesentlichen Konzepte der Informatik auf einem vertieften wissenschaftlichen Niveau mit ihren mathematisch-formalen Grundlagen: Ereignis-Log, Halbordnung und partielle Sprache, Petrinetz, Nebenläufigkeit, sequentielle und kausale Semantik eines nebenläufigen Systems, Synthese von nebenläufigen Systemen, Geschäftsprozess, Process Mining, Process Discovery. Sie können nebenläufige Systeme mittlerer Komplexität in einer geeigneten Modellierungssprache modellieren, sowie zu einem gegebenen nebenläufigen System verschiedene Verhaltensmodelle zur Analyse und Simulation generieren, bewerten und ineinander überführen. Sie kennen die verschiedenen Process Mining Anwendungsfälle mit dazu passenden Lösungstechniken und Qualitätskriterien zu deren Bewertung. Teilnehmer verfügen über eine fortgeschrittene mathematisch formale Methodik für die Analyse und Formalisierung komplexer Process Discovery Probleme. Sie sind in der Lage, geeignete Algorithmen zu deren Lösung bzgl. qualitativer und quantitativer Kriterien sicher begründet auszuwählen, sowie diese anzuwenden und die Ergebnisse zu interpretieren und zu bewerten.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen, konzeptionellen und algorithmischen Denken; Eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Verständliche Präsentation von Ergebnissen; Qualitätsbewußtsein; Akribie; Grundsätze guter wissenschaftlicher Praxis; Wissenschaftliche Methodik;</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 45 Std. Vorlesung (Präsenzstudium) 15 Std. Übung (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>		
Voraussetzungen: keine		ECTS/LP-Bedingungen: Bestehen der Modulprüfung
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Process Mining (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch / Englisch SWS: 3</p>
<p>Inhalte: Die Veranstaltung gibt einen fundierten Überblick über das Forschungsgebiet Process Mining:</p> <ul style="list-style-type: none"> - Modelling Techniques: Workflow Nets, Partial Languages - Discovering algorithms - Conformance Checking - Process Enhancement - The PROM Framework

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

Modulteil: Process Mining (Übung)

Lehrformen: Übung

Sprache: Deutsch / Englisch

SWS: 1

Prüfung

Process Mining (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0033: Modellgetriebene Softwareentwicklung		6 ECTS/LP
Version 2.0.0 (seit WS14/15) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Modellgetriebene Softwareentwicklung oder Model Driven Software Development (MDSD) befasst sich mit der Effizienzsteigerung in der Softwareherstellung durch Automatisierung und Wiederverwendung. Die Teilnehmer lernen in der Veranstaltung Methoden zur modellgetriebenen Entwicklung von Softwaresysteme anzuwenden und zu vergleichen. Sie entwickeln dabei vertiefte, fachspezifische Lösungskonzepte für MDSD. Sie können aktuelle Technologien und Standards für MDSD bewerten und analysieren deren Anwendbarkeit in praxisrelevanten Aufgabenstellungen. Die Teilnehmer bauen Fertigkeiten zur Analyse und Strukturierung komplexer Informatikproblemstellungen bei der Generierung von Infrastrukturcode, Subsysteme, Konfigurationen oder ganze Anwendungen aus Modellen auf. Dabei entwickeln sie Fertigkeiten zum logischen, analytischen und konzeptionellen Denken und können Problemlösungen systematisch entwickeln und beurteilen.</p> <p>Schlüsselqualifikation: Fachübergreifende Kenntnisse; Kompetenz zur Vernetzung unterschiedlicher Fachgebiete; Team- und Kommunikationsfähigkeit; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein; Kennenlernen der Abläufe und Prozesse im Anwendungsumfeld der Informatik; Formale quantitative Grundlagen kennen und verstehen; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 30 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 5	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Modellgetriebene Softwareentwicklung (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 3		
Inhalte: Modellgetriebene Softwareentwicklung oder Model Driven Software Development (MDSD) befasst sich mit der Effizienzsteigerung in der Softwareherstellung durch Automatisierung und Wiederverwendung. Dabei werden Infrastrukturcode, Subsysteme, Konfigurationen oder ganze Anwendungen aus Modellen generiert.		
<p>Literatur:</p> <ul style="list-style-type: none"> • Folien • Pohl et al. Software Product Line Engineering: Foundations, Principles, and Techniques • Kleppe et al: MDA explained • Hitz et al: UML@Work • weitere Literatur in der Vorlesung zu speziellen Themen 		

Modulteil: Modellgetriebene Softwareentwicklung (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Modellgetriebene Softwareentwicklung

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0034: Softwarearchitekturen und Enterprise Architecture Management		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Die Studierenden sind in der Lage SW-Architekturen zu erstellen (K3), zu bewerten (K6) und zu dokumentieren. Dafür können sie fachliche Lösungskonzepte in Modelle überführen und kennen Methoden zur Entwicklung solcher Abstraktionen und Architekturen. Sie können Vor- und Nachteile von Entwurfsalternativen beschreiben (K4) und können diese im jeweiligen Anwendungszusammenhang bewerten (K6). Probleme können dabei eigenständig erkannt (K4) und Lösungen systematisch entworfen (K5) und realisiert (K3) werden. Weiterhin haben sie Fertigkeiten für die Analyse und Strukturierung der Problemstellungen in Unternehmensarchitekturen entwickelt und kennen die Konzepte und Vorgehensweisen für die Erstellung solcher Architekturen. Die Studierenden können praxisrelevanten Fragestellungen in Unternehmensarchitekturen benennen (K1). Sie können geeignete Methoden zur Architekturerstellung und -bewertung auswählen und sicher anwenden. Die Studenten kennen Modellierungssprachen und Patterns zur Erstellung von Software- und Unternehmensarchitekturen. Sie haben die Kompetenz bedeutende technische Entwicklungen zu erkennen.</p> <p>Schlüsselqualifikation: Kompetenz zur Vernetzung unterschiedlicher Fachgebiete; Team- und Kommunikationsfähigkeit; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Praxiserfahrung und Berufsbefähigung</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 45 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
<p>Voraussetzungen: Frühere Veranstaltung "Softwarearchitekturen und Technologien für eingebettete Systeme" darf nicht belegt worden sein wegen Überschneidungen.</p>		
<p>Angebotshäufigkeit: jedes Wintersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 5</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Softwarearchitekturen und Enterprise Architecture Management (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 3</p>		
<p>Inhalte: Der Vorlesungsinhalt umfasst Patterns, Modellierungstechniken und die Evaluation von Softwarearchitekturen. Weiterhin wird auf den Bereich des Enterprise Architecture Managements eingegangen.</p>		
<p>Literatur:</p> <ul style="list-style-type: none"> • Bass et al: Software Architecture in Practice • Clements et al: Documenting Software Architectures • Clements et al: Evaluation of Software Architectures • Kopetz: Real-Time Systems 		

Zugeordnete Lehrveranstaltungen:

Softwarearchitekturen und EAM (Vorlesung)

Inhalte: - Was sind SW-Architekturen? - Pattern und Muster für SW-Architekturen - Modellierung von SW-Architekturen - Evaluation von SW-Architekturen - Eingebettete Systeme: Definitionen, Anforderungsanalyse, Modellierung, Architektur - Software-Qualität: Definitionen und Standards, Funktionstest, Überdeckungsmaße, HiL-, Integrations- und Abnahmetests, Verifikation und Validierung, Architecture Design and Reliability - Enterprise Architecture Management: Methoden, Frameworks, Tools

Modulteil: Softwarearchitekturen und Enterprise Architecture Management (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Softwarearchitekturen und EAM (Übung)

Der Übungsbetrieb soll die Inhalte der Vorlesung veranschaulichen und wird aus der gruppenweisen Vorstellung praktischer Beispiele von Architekturen und Frameworks bestehen. Die konkreten Themen folgen.

Prüfung

Softwarearchitekturen und Enterprise Architecture Management (mündl. Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0037: Praktikum Automotive Software Engineering		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Praktikum Automotive Software Engineering verstehen die Studierenden praxisnaher Problemstellungen höherer Komplexität im Bereich der Entwicklung und Absicherung von Fahrassistenzsystemen mit aktuellen Methoden und Tools der modellbasierten Entwicklung eingebetteter Systeme. Die Studierenden erlangen tiefergehende fachspezifische als auch fächerübergreifende Kenntnisse und Fähigkeiten, beispielsweise aus der Regelungstechnik, Fahrphysik und Mathematik. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren, zu präsentieren und verständlich zu dokumentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; Verständliche, sichere und überzeugende Präsentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein; Kommunikationsfähigkeit; Fertigkeit der Zusammenarbeit in Teams und Verstehen von Teamprozessen; Projektmanagementfähigkeiten</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 90 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p>Voraussetzungen: 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</p>		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Praktikum Automotive Software Engineering (MA) Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		

Inhalte:

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.

Literatur:

abhängig vom Thema

Zugeordnete Lehrveranstaltungen:

Praktikum zu Automotive Software Engineering (Praktikum)

Im Automotive-Praktikum lernen die Teilnehmer/innen wie Software für Automotive-Anwendungen modellbasiert entwickelt, simuliert und analysiert werden kann. In einem Einführungskurs werden die notwendigen Grundlagen anhand von eigens dafür konzipierten Tutorials erlernt. Die Teilnehmer/innen lernen dabei u.a. die im Automotive-Umfeld häufig eingesetzte Modellierungswerkzeug-Kombination „MATHLAB/Simulink“ sowie das graphische Simulations- und Analyse-Tool „CarMaker“ kennen. Im Anschluss an den Einführungskurs werden in Gruppen von drei bis vier Teilnehmer/innen Übungsaufgaben zur Vertiefung der im Einführungskurs erlernten Kenntnisse bearbeitet. Dies garantiert eine optimale Vorbereitung an die anschließende Projektbearbeitung. Im Rahmen des Projektes werden in Teamarbeit Fahrerassistenzsysteme oder Teilaspekte des autonomen Fahrens modelliert, implementiert und getestet. In einer abschließenden Präsentation werden die Ergebnisse vorgestellt und diskutiert. Die Arbeitsunterlagen (Tutoria ... (weiter siehe Digicampus)

Prüfung

Praktikum Automotive Software Engineering (MA)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0038: Praktikum Avionic Software Engineering	6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer	
<p>Inhalte:</p> <p>Die Teilnehmer im Avionik-Praktikum erlernen, wie Software für komplexe Avionik-Systeme entwickelt wird. In kleinen Gruppen wird von den Studenten ein einfacher Autopilot für ein fliegendes System umgesetzt und in einer Simulationsumgebung getestet. Die so entwickelte Steuerung soll anschließend auf einem realen Quadrocopter zum Einsatz kommen. Die Studierenden erarbeiten anhand der Aufgabenstellung eine Software-Architektur, und realisieren diese innerhalb eines Frameworks zur Anbindung an die Simulationsumgebung (X-Plane).</p> <p>In einer mehrtägigen Einführungs-Blockveranstaltung erwerben die Teilnehmer die nötigen Grundkenntnisse über die Entwicklung zuverlässiger Avionik-Systeme und erhalten einen Überblick über die für dieses Praktikum verwendeten Konzepte und Technologien:</p> <ul style="list-style-type: none"> • Techniken zur Entwicklung sicherheitskritischer Systeme • Qualitätssicherung in der Software-Entwicklung durch den Einsatz geeigneter Werkzeuge • Einführung in die Steuerung fliegender Systeme und Navigation • Einführung in die Physik fliegender Systeme • Komponentenbasierte Software-Entwicklung mit Java und OSGi • Echtzeitfähige Software in Java gemäß der RTSJ-Spezifikation <p>Das Praktikum wird in den Semesterferien angeboten und besteht aus dem theoretischen Teil als Blockveranstaltung und der anschließenden mehrwöchigen selbstständigen Umsetzung der Praktikumsaufgabe durch die Studierenden.</p> <p>Die erforderlichen Tätigkeiten sind:</p> <ul style="list-style-type: none"> • Erstellung einer geeigneten Software-Architektur und -Design • Implementierung eines grundlegenden Autopilots innerhalb des vorgebenen Frameworks in Java und OSGi • Entwicklung von Software-Komponenten zur Steuerung der Flughardware • Überprüfung der funktionalen Korrektheit durch Unit- und Integrationstests 	
<p>Lernziele/Kompetenzen:</p> <p>Nach der Teilnahme am Praktikum Avionic Software Engineering verfügen die Studierenden über detailliertes und aktuelles Wissen auf dem Gebiet der Softwareentwicklung komplexer Systeme im Bereich der Luftfahrt. Sie können Problemstellungen und Ergebnisse des Gebiets präzise beschreiben und vergleichen. Darüber hinaus verstehen die Studierenden tiefere Konzepte, Methoden, Verfahren, Techniken und Technologien aus dem genannten Gebiet, um eigenere Lösungsansätze zu planen und die erlernten Kompetenzen bei der Umsetzung aktiv anzuwenden. Durch die Lösung der praktischen Aufgabenstellung in Gruppen entwickeln sie die Team- und Kommunikationsfähigkeit, um Problemstellungen auf dem Gebiet zu erörtern, Fragen und Zwischenergebnisse zu diskutieren und die gemeinsam ausgearbeiteten Lösungen zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Recherche in (auch englischsprachlichen) Lehrbüchern, Handbüchern und Dokumentationen; Verständliche, sichere und überzeugende Präsentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein; Kommunikationsfähigkeit; Projektmanagementfähigkeiten; Fertigkeit der Zusammenarbeit in Teams und Verstehen von Teamprozessen; Kenntnisse von praxisrelevanten Aufgabenstellungen</p>	
<p>Arbeitsaufwand:</p> <p>Gesamt: 180 Std.</p> <p>90 Std. Praktikum (Präsenzstudium)</p> <p>90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>	
<p>Voraussetzungen:</p> <p>Grundkenntnisse im Bereich Software Engineering; Programmiererfahrung in Java; Interesse an Avionik-Systemen; Keine Erfahrung mit OSGi erforderlich</p>	

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		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Moduleile
<p>Moduleil: Praktikum Avionic Software Engineering</p> <p>Lehrformen: Praktikum</p> <p>Sprache: Deutsch</p> <p>SWS: 6</p>
<p>Inhalte:</p> <p>Die Teilnehmer im Avionik-Praktikum erlernen, wie Software für komplexe Avionik-Systeme entwickelt wird. In kleinen Gruppen wird von den Studenten ein einfacher Autopilot für ein fliegendes System umgesetzt und in einer Simulationsumgebung getestet. Die so entwickelte Steuerung soll anschließend auf einem realen Quadrocopter zum Einsatz kommen. Die Studierenden erarbeiten anhand der Aufgabenstellung eine Software-Architektur, und realisieren diese innerhalb eines Frameworks zur Anbindung an die Simulationsumgebung (X-Plane).</p> <p>In einer mehrtägigen Einführungs-Blockveranstaltung erwerben die Teilnehmer die nötigen Grundkenntnisse über die Entwicklung zuverlässiger Avionik-Systeme und erhalten einen Überblick über die für dieses Praktikum verwendeten Konzepte und Technologien:</p> <ul style="list-style-type: none"> • Techniken zur Entwicklung sicherheitskritischer Systeme • Qualitätssicherung in der Software-Entwicklung durch den Einsatz geeigneter Werkzeuge • Einführung in die Steuerung fliegender Systeme und Navigation • Einführung in die Physik fliegender Systeme • Komponentenbasierte Software-Entwicklung mit Java und OSGi • Echtzeitfähige Software in Java gemäß der RTSJ-Spezifikation <p>Das Praktikum wird in den Semesterferien angeboten und besteht aus dem theoretischen Teil als Blockveranstaltung und der anschließenden mehrwöchigen selbstständigen Umsetzung der Praktikumsaufgabe durch die Studierenden.</p> <p>Die erforderlichen Tätigkeiten sind:</p> <ul style="list-style-type: none"> • Erstellung einer geeigneten Software-Architektur und -Design • Implementierung eines grundlegenden Autopiloten innerhalb des vorgebenen Frameworks in Java und OSGi • Entwicklung von Software-Komponenten zur Steuerung der Flughardware • Überprüfung der funktionalen Korrektheit durch Unit- und Integrationstests
<p>Literatur:</p> <p>abhängig vom Thema</p>
<p>Zugeordnete Lehrveranstaltungen:</p> <p>Praktikum Avionic Software Engineering (Praktikum)</p> <p>Die Teilnehmer im Avionik-Praktikum erlernen, wie Software für komplexe Avionik-Systeme entwickelt wird. In kleinen Gruppen wird von den Studenten ein einfacher Autopilot für ein fliegendes System umgesetzt und in einer Simulationsumgebung getestet. Die so entwickelte Steuerung soll anschließend auf einem realen Quadrocopter zum Einsatz kommen. Die Studierenden erarbeiten anhand der Aufgabenstellung eine Software-Architektur, und realisieren diese innerhalb eines Frameworks zur Anbindung an die Simulationsumgebung (X-Plane). In einer mehrtägigen Einführungs-Blockveranstaltung erwerben die Teilnehmer die nötigen Grundkenntnisse über</p>

die Entwicklung zuverlässiger Avionik-Systeme und erhalten einen Überblick über die für dieses Praktikum verwendeten Konzepte und Technologien: • Techniken zur Entwicklung sicherheitskritischer Systeme • Qualitätssicherung in der Software-Entwicklung durch den Einsatz geeigneter Werkzeuge • Einführung in die Steuerung fliegender Systeme und Navigation • Einf
... (weiter siehe Digicampus)

Prüfung

Praktikum Avionic Software Engineering

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0066: Organic Computing II		5 ECTS/LP
Version 1.2.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen: Teilnehmer sind in der Lage, wesentliche Konzepte und Methoden des Organic Computing (OC) wiederzugeben. Dazu zählen unter anderem: Selbst-Organisation, Selbst-Adaption, Robustheit, Flexibilität, Observer/Controller-Architekturen (O/C), Selbst-X Eigenschaften, Extended Classifier Systems (XCS), Genetische Algorithmen (GA), Partikel Swarm Optimization, Influence Detection und Trust. Sie können zudem begründen, wieso es sinnvoll ist OC-Systeme zu betrachten und praxisrelevante Beispiele nennen, in welchen OC-Techniken angewendet werden sollten. Die Teilnehmer sind befähigt, größere Softwaresysteme mit Hilfe der O/C-Architektur zu entwickeln, sowie diese mit passenden OC-Techniken zu befüllen und sich, falls erforderlich, neue OC-Techniken anzueignen ein gegebenes Problem zu lösen. Die Studenten haben die Fähigkeit zum Lösen von praktischen Aufgaben mit Hilfe des XCS, dem Swarming und einem GA. Sie können zudem ein Konzept für ein OC-System ausarbeiten und beurteilen, sowie wissenschaftliche Bewertungen im Bezug auf OC-Systeme durchführen.</p> <p>Schlüsselqualifikationen: Fähigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen, Fähigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten., Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen, Fertigkeit der Zusammenarbeit in Teams.</p>		
<p>Arbeitsaufwand: 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)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Organic Computing II (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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 naturanaloger 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>		

Literatur:

- Folien
- Müller-Schloer und Tomforde: Organic Computing - Technical Systems for Survival in the Real World, Springer International Publishing, 2018, ISBN 978-3-319-68477-2
- Müller-Schloer et al.: Organic Computing - A Paradigm Shift for Complex Systems, Birkhäuser Verlag, Basel, 2011, ISBN 978-3034801294
- Würtz (ed.): Organic Computing (Understanding Complex Systems), SpringerVerlag Berlin, 2008, ISBN 978-3540776567
- Mitchell: Machine Learning, The McGraw-Hill Companies, 1997, ISBN 978-0071154673
- Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989, ISBN 978-0201157673
- Michalewicz, Fogel: How to Solve it: Modern Heuristics, Springer Verlag Berlin, 2004, ISBN 978-3540224945
- Tomforde: Runtime Adaptation of Technical Systems, Südwestdeutscher Verlag für Hochschulschriften, 2012, ISBN 978-3838131337

Modulteil: Organic Computing II (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

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.

Modul INF-0077: Suchmaschinen		8 ECTS/LP
Version 1.2.0 (seit SoSe14) Modulverantwortliche/r: PD Dr. Markus Endres		
<p>Lernziele/Kompetenzen: 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. Hierfür sind vertiefende Kenntnisse im Bereich der Suchmaschinentechologie notwendig. Darunter zählen z.B. die Architektur einer Suchmaschine, Web Crawler, Textprocessing, Indexierung, verschiedene Retrieval Modelle, die Evaluation von Suchalgorithmen, Clusterverfahren, Link Analyse, uvm.</p> <p>Die Studierenden erwerben somit Kenntnisse verschiedener Suchmaschinenaspekte und können die Vor- und Nachteile in praxisrelevanten Aufgabenstellungen abwägen und beurteilen. Zudem können die Studierenden nach der Veranstaltung wissenschaftlich aussagekräftige Bewertungen erstellen und diese in praxisrelevante Lösungsstrategien umsetzen. In der Veranstaltung werden zudem relevante Themen aus der Forschung anhand aktueller Veröffentlichungen angesprochen, so dass die Studierenden die Fähigkeit erwerben, ihr vorhandenes Wissen selbständig zu erweitern.</p>		
<p>Schlüsselqualifikationen: Fachspezifische Vertiefung; Umsetzen fachlicher Lösungskonzepte in Programme und Modelle; Kenntnisse der Vor-/Nachteile von Entwurfsalternativen und Bewertung im jeweiligen Anwendungszusammenhang; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit, mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen anzustellen; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen;</p>		
<p>Arbeitsaufwand: 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>Voraussetzungen: Modul Datenbanksysteme (INF-0073) - empfohlen</p>		
Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Suchmaschinen (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4</p>		
<p>Inhalte: Die Vorlesung behandelt grundlegende Konzepte von Suchmaschinen, Volltext-Suche, SQL-Suchmaschinen und Präferenz-Suchmaschinen (Preference SQL) sowie deren Implementierung.</p>		

Literatur:

- M. Levene: An Introduction to Search Engines and Web Navigation
- R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval
- I. H. Witten, M. Gori, T. Numerico: Web Dragons
- W. Kießling: Foundations of Preferences in Database Systems
- W. Kießling: Preference Queries with SV-Semantics

Modulteil: Suchmaschinen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Suchmaschinen (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0078: Datenbankprogrammierung (Oracle)		5 ECTS/LP
Version 1.2.0 (seit SoSe14) Modulverantwortliche/r: PD Dr. Markus Endres		
<p>Lernziele/Kompetenzen: Nach der Teilnahme an der Veranstaltung sind die Studierenden in der Lage vertiefte Oracle-Kenntnisse in praxisrelevanten Aufgaben anzuwenden. Hierzu ist erweitertes Wissen im Bereich DB-System-Architektur, DB-Entwurfstheorie, Administration, erweitertes SQL, PL/SQL-Programmierung, Baumstrukturen, XML und OLAP notwendig. Die Studierenden erlangen eine fachspezifische Vertiefung im Bereich der Datenbanksysteme und somit die Fertigkeit zur Analyse und Strukturierung von Problemstellungen. Darüber hinaus können die Studierenden diese komplexen, praxisrelevanten Aufgabenstellungen 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 unter praxisnahen Randbedingungen verfassen.</p> <p>Schlüsselqualifikationen: Umsetzen fachlicher Lösungskonzepte in Programm eund Modelle, Fachspezifische Vertiefung; Fachübergreifende Kenntnisse; Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen; Kenntnisse der Vor-/Nachteile von Entwurfsalternativen und Bewertung im jeweiligen Zusammenhang; Auswahl und sichere Anwendung geeigneter Methoden; Kenntnisse von praxisrelevanten Aufgabenstellungen; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen;</p>		
<p>Arbeitsaufwand: 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. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)</p>		
Voraussetzungen: Modul Datenbanksysteme (INF-0073) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Datenbankprogrammierung (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: 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.		
Literatur: <ul style="list-style-type: none"> • R. Elmasri, S. Navathe: Fundamentals of Database Systems • S. Melton: Understanding the New SQL: A Complete Guide • Oracle 11g Online-Dokumentation 		

Zugeordnete Lehrveranstaltungen:

Datenbankprogrammierung (Oracle) (Vorlesung)

Datenbanken haben sich als allgegenwärtiges Werkzeug im öffentlichen, wissenschaftlichen und wirtschaftlichen Leben etabliert. Diese Vorlesung wendet sich an DB-Interessierte, die ihre vorhandenen Kenntnisse aus einer grundlegenden Datenbankvorlesung mit Hilfe von Oracle vertiefen bzw. erweitern wollen. Daher ist die Vorlesung insbesondere für Studierende geeignet, die ihren Schwerpunkt im Bereich Datenbanken- und Informationssysteme setzen bzw. vertiefte praktische Kenntnisse erwerben wollen.

Modulteil: Datenbankprogrammierung (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Datenbankprogrammierung (Oracle) (Übung)

Prüfung

Datenbankprogrammierung (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

Modul INF-0117: Funktionale Modellierung für Geoinformationssysteme		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller Prof. Dr. Sabine Timpf		
Lernziele/Kompetenzen: Die Studierenden verfügen über ein tiefes 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.		
Schlüsselqualifikationen: analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis		
Arbeitsaufwand: 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)		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Funktionale Modellierung für Geoinformationssysteme (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: 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		
Literatur: <ul style="list-style-type: none"> • Eigenes Skriptum • B O'Sullivan, D. Stewart, J. Goerzen: Real World Haskell, O'Reilly 2008 • M.Worboys, M. Duckham: GIS - A computing perspective, Routledge 2004 		
Modulteil: Funktionale Modellierung für Geoinformationssysteme (Übung)		
Lehrformen: Übung		
Sprache: Deutsch		
SWS: 2		

Prüfung

Funktionale Modellierung für Geoinformationssysteme (Klausur)

Klausur / Prüfungsdauer: 120 Minuten

Modul INF-0129: Softwaretechnik II		8 ECTS/LP
Version 2.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen:</p> <p>Fachbezogene Kompetenzen:</p> <p>Nach dem erfolgreichen Besuch des Moduls sind die Teilnehmer in der Lage, fortgeschrittene Verfahren der agilen Softwareentwicklung, des Requirements Engineerings, des Testens und des Refactorings von Softwaresystemen anzuwenden und miteinander zu verbinden. Sie können die Anwendbarkeit der verschiedenen Verfahren und Methoden situationsspezifisch beurteilen.</p> <p>Die Teilnehmer verstehen die Aufgaben des Requirements Engineering und sind in der Lage, wesentliche Methoden der Requirements-Erfassung und Dokumentation (Use-Cases, Satzschablonen, formale Modelle) anzuwenden und die Eignung der verschiedenen Dokumentationsformen im Projektkontext zu bewerten. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken und können so systematisch komplexe Kundenanforderungen analysieren und können diese formulieren und in geeigneten Modellen darstellen.</p> <p>Die Studierenden verstehen einen agilen Entwicklungsprozess und die Basisprinzipien agiler Vorgehensweisen. Sie können den agilen Entwicklungsprozess auf Projekte übertragen und seine Anwendbarkeit im Projektkontext beurteilen. Sie kennen agile Praktiken und deren Wechselwirkungen und können die Praktiken einsetzen. Sie verstehen die Aufgaben der beteiligten Rollen und die Bausteine des Entwicklungsprozesses.</p> <p>Die Studierenden verstehen die Aufgaben des Testens und die Teilschritte des Testprozesses. Sie kennen Verfahren zur Bestimmung von Testfällen und zur Bewertung der Adäquatheit von Testsuiten und können diese anwenden.</p> <p>Die Studierenden 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>Die Studierenden erwerben insgesamt Kenntnisse in Methoden zur Entwicklung großer Softwaresysteme und der Konstruktion von Abstraktionen sowie die Fähigkeit zu deren Anwendung im Projektkontext. Sie können alternative Vorgehensweisen wissenschaftlich fundiert bewerten und für ein Projekt geeignete auswählen.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Abstraktionsfähigkeit • Moderieren fachlicher Sitzungen • Erlernen des eigenständigen Arbeitens mit Lehrbüchern und Arbeit in selbstorganisierten Teams • Kenntnisse von praxisrelevanten Aufgabenstellungen 		
<p>Arbeitsaufwand:</p> <p>Gesamt: 240 Std.</p> <p>90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> <p>30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> <p>60 Std. Vorlesung (Präsenzstudium)</p>		
<p>Voraussetzungen:</p> <p>Programmierkenntnisse in Java (empfohlen)</p> <p>Modul Softwaretechnik (INF-0120) - empfohlen</p>		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester

SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Moduleile		
Modulteil: Softwaretechnik II (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte:		
<p>Agile Softwareentwicklung:</p> <ul style="list-style-type: none"> • Entwicklungsmethoden (Scrum) • Agile Praktiken • Agile Werte, Prinzipien und Methoden <p>Refactoring</p> <ul style="list-style-type: none"> • Code Smells • Prinzipien des objektorientierten Designs • Wichtige Refactorings <p>Testen</p> <ul style="list-style-type: none"> • Testprozess und Ziele des Testens • Testarten • Methoden zur Testfallgewinnung • Adäquatheitskriterien beim Blackbox- und Whitebox-Testen <p>Requirements Engineering</p> <ul style="list-style-type: none"> • Aufgaben, Begriffe und Artefakte • RE-Prozess • Techniken zur Requirements-Elicitation, -Analyse und -Dokumentation • Qualitätskriterien für Software-Requirements 		
Literatur:		
<ul style="list-style-type: none"> • 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 		
Zugeordnete Lehrveranstaltungen:		
Softwaretechnik 2 (Vorlesung)		
Modulteil: Softwaretechnik II (Übung)		
Lehrformen: Übung		
Sprache: Deutsch		
SWS: 4		
Zugeordnete Lehrveranstaltungen:		
Übung zu Softwaretechnik 2 (Übung)		

Prüfung

Softwaretechnik II Klausur

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0130: Formale Methoden im Software Engineering		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
Lernziele/Kompetenzen: Fachbezogene Kompetenzen: Die Studierenden können mathematisch-formale Methoden für die Programmverifikation, speziell bei sicherheitskritischer Software einsetzen. Durch das Verständnis von Kalkülen trainieren sie die Fertigkeit zum logischen, konzeptuellen und analytischen Denken. Sie können Spezifikationen von Datenstrukturen entwerfen und deren Eigenschaften formal beweisen. Sie sind in der Lage, funktionale Eigenschaften von sequentiellen und nebenläufigen Programmen zu formulieren und dafür Beweise zu erstellen. Sie erwerben die Fertigkeit zur abstrakten, modularen Modellierung von Softwaresystemen und zur systematischen Entwicklung von korrekten Programmen aus diesen Modellen. Sie erwerben dadurch die Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen und können mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen abgeben. Sie erwerben Fähigkeiten, um Beiträge zur Wissenschaft zu leisten.		
Schlüsselqualifikationen: <ul style="list-style-type: none"> • Training des logischen Denkens • Analytisch-methodische Kompetenz • Erwerb von Abstraktionsfähigkeiten • Fähigkeiten zur Modellbildung für Softwaresysteme 		
Arbeitsaufwand: 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)		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Formale Methoden im Software Engineering (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2		
Inhalte: 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		

Literatur:

- Sperschneider, Antoniou: Logic: A Foundation for Computer Science, Addison Wesley 1991
- Loeckx, Ehrich, Wolf: Specification of Abstract Data Types, Wiley 1996
- Ausführliche Dokumentation
- Folienhandout

Modulteil: Formale Methoden im Software Engineering (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Formale Methoden im Software Engineering (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Modul INF-0131: Software- und Systemsicherheit		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen:</p> <p>Fachbezogene Kompetenzen:</p> <p>Nach der erfolgreichen Teilnahme an der Lehrveranstaltung kennen Studierende die Grundlagen, Aufbau, und typische Schwächen kryptographischer Protokolle und praxisrelevanter, sicherheitskritischer Systeme, können diese verständlich und sicher darstellen, und verstehen die Grundprinzipien deren Sicherheitsanalyse.</p> <p>Sie können Bedrohungsanalyse durchführen, kryptographische Protokolle entwickeln, Chipkarten programmieren, sicherheitskritische Systeme in Teams entwerfen, und ihre Ergebnisse dokumentieren.</p> <p>Sie können systematisch Bedrohungen für Softwaresysteme analysieren und deren Risiken bewerten.</p> <p>Sie können einen modellgetriebenen Entwicklungsprozess für sicherheitskritische Systeme anwenden.</p> <p>Sie sind in der Lage, bedeutende technische Entwicklungen zu erkennen und fachliche Lösungskonzepte in Programme umzusetzen. Sie haben Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Abstraktionsfähigkeiten • Grundsätze guter wissenschaftlicher Praxis • Mathematisch-formale Grundlagen • Quantitative Aspekte der Informatik • Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen • Fähigkeit, vorhandenes Wissen selbstständig zu erweitern • Qualitätsbewusstsein und Akribie 		
<p>Arbeitsaufwand:</p> <p>Gesamt: 240 Std.</p> <p>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>60 Std. Übung (Präsenzstudium)</p>		
Voraussetzungen:		
keine		
Angebotshäufigkeit: unregelmäßig (i. d. R. im WS)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Software- und Systemsicherheit (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p>		

Inhalte:

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.

Literatur:

- Schneier: Applied Cryptography, Wiley and Sons, 1996 (2nd edition)
- Anderson, Needham: Programming Satan's Computer, in: Computer Science Today, Springer LNCS 1000, 1995
- 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
- Folienhandout, Spezifikationen und APIs

Modulteil: Software- und Systemsicherheit (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Software- und Systemsicherheit (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Modul INF-0133: Selbstorganisierende, adaptive Systeme		8 ECTS/LP
Version 2.0.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen: Die Studierenden haben vertiefte Fachkenntnisse über 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. Die Studierenden sind in der Lage, adaptive Systeme adäquat zu modellieren und dokumentieren.</p> <p>Außerdem kennen die Studierenden praxisrelevante Fragestellungen.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten • Grundsätze guter wissenschaftlicher Praxis 		
<p>Arbeitsaufwand: 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 anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Selbstorganisierende, adaptive Systeme (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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>		

Literatur:

- Gleick: Chaos: Making a New Science, Penguin 2008
- Strogatz: Sync : The Emerging Science of Spontaneous Order, Hyperion 2003
- Miller, Page: Complex Adaptive Systems: An Introduction to Computational Models of Social Life, Princeton University Press 2007
- Dawkins: The Selfish Gene, Oxford University Press, 3rd Revised Edition
- Wolfram: A New Kind of Science, Wolfram Media Inc. 2002
- von Neumann, Morgenstern: Theory of Games and Economic Behavior, 2004
- Folienhandout

Zugeordnete Lehrveranstaltungen:

Selbst-organisierende, adaptive Systeme (Vorlesung)

"Selbst-organisierende, adaptive Systeme" beschäftigt sich mit theoretischen Grundlagen, die für die Entwicklung von offenen Multiagentensystemen nötig sind. Dabei folgt die Vorlesung vor allem dem Aufbau von <http://www.masfoundations.org/> und behandelt Spieltheorie, Mechanism Design und (verteilte) Constraint-Optimierung. Sie richtet sich vor allem an den Anwendungsfällen des Lehrstuhls "Verteiltes Energiemanagement" (<http://www.focas-reading-room.eu/robust-optimization-in-self-organizing-power-management/>) sowie flexible Produktion (<https://www.isse.uni-augsburg.de/projects/reif/completed/save-orca/>) und Multi-Roboter Arbeitszuweisung im Ensemble Programming (<https://www.isse.uni-augsburg.de/projects/reif/robot-ensembles/>) aus.

Modulteil: Selbstorganisierende, adaptive Systeme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Selbst-organisierende, adaptive Systeme (Übung)

Prüfung

Selbstorganisierende, adaptive Systeme (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0189: Qualitätssicherung im Software Engineering		5 ECTS/LP
Version 2.0.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen: Nach erfolgreicher Absolvierung der Lehrveranstaltung verstehen Studierende die Grundtechniken der Qualitätssicherung und der Problemstellungen des Testens in der Praxis. Sie sind in der Lage, SW-Module zu spezifizieren und kennen die wesentlichen Testtechniken und deren Einsatzzwecke im Software Engineering. Sie können Testkonzepte für große Softwaresysteme entwerfen, modellieren und anwenden. Die Studierenden sind für das Thema Qualität im Software Engineering sensibilisiert und können verschiedene Qualitätskriterien/-metriken kritisch hinterfragen und bewerten.</p> <p>Des Weiteren kennen und verstehen sie die Prinzipien von konstruktiven Qualitätssicherungstechniken und -praktiken.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Training des logischen Denkens • Analytisch-methodische Kompetenz • Abwägen von Vorgehensweisen • Erwerb von Abstraktionsfähigkeiten • Verbesserung der eigenen Softwareentwicklungskompetenz 		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Qualitätssicherung im Software Engineering (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		

Inhalte:

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.

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 (Mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0213: Parallele und Verteilte Datenbanksysteme		5 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: PD Dr. Markus Endres		
<p>Lernziele/Kompetenzen: 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 im Detail zu verstehen, zu bewerten und anzuwenden. Darüber hinaus können die Studierenden weiterführende komplexe Problemstellungen auf dem Gebiet Datenbanken, insbesondere unter Verwendung von parallelen und verteilten Algorithmen, analysieren und lösen. Die Vorlesung vermittelt zudem die Fertigkeit zum logischen, analytischen und konzeptionellen Denken. Bei den in der Veranstaltung anfallenden praxisnahen Programmieraufgaben ist eine sichere und überzeugende Darstellung von Ideen und Konzepten sowie die Zusammenarbeit in Teams entscheidend.</p> <p>Schlüsselqualifikationen: Fachspezifische Vertiefung; Kenntnisse der Vor-/Nachteile von Entwurfalternativen und deren Bewertung im jeweiligen Anwendungszusammenhang; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen; Fähigkeit, vorhandenes Wissen selbständig zu erweitern;</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 3.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Parallele und Verteilte Datenbanksysteme (Vorlesung)		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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>Literatur:</p> <ul style="list-style-type: none"> • - Taniar et al.: High-Performance Parallel Database Processing and Grid Databases • - E. Rahm: Mehrrechner-Datenbanksysteme • - M. T. Özsu, P. Valduriez: Principles of Distributed Database Systems • - P. Dada: Verteilte Datenbanken und Client/Server-Systeme • - S. Conrad: Förderierte Datenbanksysteme • - S. K. Rahimi, F. S. Haug: Distributed Database Management Systems - A Practical Approach 		

Modulteil: Parallele und Verteilte Datenbanksysteme (Übung)

Sprache: Deutsch

SWS: 2

Prüfung

Parallele und Verteilte Datenbanksysteme (mdl. Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0217: Praktikum Autonomes Fahren		10 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Praktikum Autonomes Fahren verstehen es die Studierenden praxisnahe Problemstellungen hoher Komplexität im Bereich der Konzeptionierung, Entwicklung und Absicherung von hochautomatisierten/ autonomen Fahrzeugen mit aktuellen Methoden und Tools der modellbasierten Entwicklung zu lösen. Die Studierenden erlangen tiefgehende fachspezifische als auch fächerübergreifende Kenntnisse und Fähigkeiten, beispielsweise aus der Hardwarenahen Informatik, dem Software Engineering, als auch der zugrundeliegenden Fahrphysik und Mathematik. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren, zu präsentieren und verständlich zu dokumentieren.</p> <p>Schlüsselqualifikation: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; Verständliche, sichere und überzeugende Präsentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein; Kommunikationsfähigkeit; Fertigkeit der Zusammenarbeit in Teams und Verstehen von Teamprozessen; Projektmanagementfähigkeiten</p>		
<p>Arbeitsaufwand: Gesamt: 300 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 150 Std. Praktikum (Präsenzstudium)</p>		
<p>Voraussetzungen: 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>		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 10	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Praktikum Autonomes Fahren Lehrformen: Praktikum Sprache: Deutsch / Englisch SWS: 10</p>		

Inhalte:

In diesem Praktikum lernen die Teilnehmer, wie verschiedene ausgewählte Teilaspekte des Autonomen Fahrens umgesetzt, simuliert und analysiert werden können.

In einem Einführungskurs werden die benötigten theoretischen Grundlagen für die Bearbeitung der Praxisaufgaben gelegt.

Darüber hinaus lernen die Teilnehmer u.a. das im Automotive-Umfeld häufig eingesetzte Entwicklungswerkzeug „ADTF“ von Elektrobit und die darin verwendete Programmiersprache C++ kennen.

Nach dem Einführungskurs sollen in Gruppen von zwei bis drei Teilnehmern mithilfe der genannten Werkzeuge autonome Fahrfunktionen umgesetzt werden.

Die entwickelten Ergebnisse werden abschließend an 1:8 Fahrzeugmodellen, deren Sensorik realen Fahrzeugen sehr nahe kommt, demonstriert und ausgewertet.

Prüfung

Praktikum Autonomes Fahren

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0233: Industrierobotik		8 ECTS/LP
Version 2.1.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
Lernziele/Kompetenzen: Die Studierenden sind in der Lage Industrieroboter zu programmieren, kennen das dafür erforderliche methodische Vorgehen und können die Eignung verschiedener Vorgehensweise bewerten. 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.		
Schlüsselqualifikationen: <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten 		
Bemerkung: Dieses Modul ersetzt das Modul "INF-0132: Software in Mechatronik und Robotik". Sofern dieses bereits belegt wurde, ist eine erneute Belegung nicht möglich.		
Arbeitsaufwand: 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)		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Industrierobotik (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2		
Inhalte: 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.		
Literatur: <ul style="list-style-type: none"> • L. Sciavicco, B. Siciliano: Modelling and Control of Robot Manipulators. Reihe: Advanced Textbooks in Control and Signal Processing. Springer 2000 (2nd Ed.) • Handbücher von KUKA • Folienhandout 		
Zugeordnete Lehrveranstaltungen:		

Industrierobotik (Vorlesung)

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

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 3 Monate

Modul INF-0235: Software für Industrie 4.0		5 ECTS/LP
Version 1.1.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
Lernziele/Kompetenzen: Die Studierenden sind in der Lage die praktischen und methodischen 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 und kennen so unterschiedliche anwendungsrelevante Disziplinen. Sie sind in der Lage Lösungskonzepte zu erstellen und dabei Entwurfsalternativen bewerten und anwenden.		
Schlüsselqualifikationen: <ul style="list-style-type: none"> • Analytisch-methodische Kompetenzen • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten 		
Arbeitsaufwand: Gesamt: 150 Std. 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) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Software für Industrie 4.0 (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: Die Vorlesung behandelt aktuelle Themen im Kontext von Industrie 4.0. Dazu zählen folgende Inhalte: <ul style="list-style-type: none"> • Einführung in die Automatisierung • Referenzarchitekturen für Industrie 4.0 • Einführung in die (mobile Service) Robotik • OPC UA • AutomationML • Data Analytics für Industrie 4.0 		
Literatur: <ul style="list-style-type: none"> • 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

Software für Industrie 4.0

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Modul INF-0248: Kollaborative Robotik		8 ECTS/LP
Version 1.1.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen: Die Studierenden sind in der Lage kollaborative Roboter zu programmieren und verstehen die regulatorischen Randbedingungen (vgl. ISO/TS 15066:2016). Sie verstehen 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. Die Studierenden können abwägen, für welche praxisrelevanten Problemstellungen kollaborative Roboter eingesetzt werden können.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten • Erwerb von Grundlagen für die Sicherheitsbewertung 		
<p>Bemerkung: Voraussetzungen (empfohlen): Software für Industrie 4.0 oder Industrierobotik oder Grundlagen der Robotik</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std.</p>		
<p>Voraussetzungen: keine</p>		<p>ECTS/LP-Bedingungen: Bestehen der Modulprüfung</p>
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Kollaborative Robotik (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 4</p>		

Inhalte:

Die Digitalisierung dringt mittlerweile in viele Bereiche des täglichen Lebens vor. Von Industriemaschinen und Robotersystemen (*Smart Factory, Industrie 4.0*) bis hin zu intelligenten Hausgeräten oder Heizungsanlagen (*Smart Home*) 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.

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.

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.

Infolgedessen liegt der thematische Fokus der Veranstaltung auf

- der Umsetzung multi-modaler Benutzerschnittstellen,
- der Gestaltung neuartiger, auch physischer Interaktionsmöglichkeiten,
- der Realisierung intelligenter Verhaltensweisen durch Techniken der Künstlichen Intelligenz bzw. des Machine Learnings und
- der Entwicklung kognitiver Fähigkeiten des Robotersystems.

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.

Zugeordnete Lehrveranstaltungen:**Übung zu Kollaborative Robotik (Übung)****Modulteil: Kollaborative Robotik (Vorlesung)**

Lehrformen: Vorlesung

Sprache: Deutsch

SWS: 2

Literatur:

- ISO/TS 15066:2016. 2016-02-15. Robots and robotic devices - Collaborative robots
- 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
- DIN EN ISO 10218-1:2012. 01.2012. Industrieroboter - Sicherheitsanforderungen - Teil 1: Roboter (ISO 10218-1:2011); Deutsche Fassung EN ISO 10218-1:2011

Zugeordnete Lehrveranstaltungen:**Kollaborative Robotik (Vorlesung)**

Der Schwerpunkt der Veranstaltung liegt auf der Entwicklung von Software für kollaborative (mobile) Manipulatoren; die grundlegende Hardware, d.h. die Roboter, mobilen Plattformen, Greifer und Sensoren, wird

bereitgestellt. 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. Infolgedessen liegt der thematische Fokus der Veranstaltung auf • der Umsetzung multi-modaler Benutzerschnittstellen, • der Gestaltung neuartiger, auch physischer Interaktionsmöglichkeiten, • der Realisierung intelligenter Verhaltensweisen durch Techniken der Künstlichen Intelligenz bzw. des Machine Learnings und • der Entwicklung kognitiver Fähigkeiten des Robotersystems. Diese Kernelemente bieten die Grundlage, um eine praktische Aufgabenstellung prototypisch umzusetzen. Die Programmierung erfolgt dab
... (weiter siehe Digicampus)

Prüfung

Kollaborative Robotik

praktische Prüfung / Prüfungsdauer: 60 Minuten

Bearbeitungsfrist: 3 Monate

Modul INF-0277: Analyzing Massive Data Sets		8 ECTS/LP
Version 1.1.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Peter Michael Fischer		
<p>Lernziele/Kompetenzen: Nach der Teilnahme an der Veranstaltung sind die Studierenden in der Lage die Konzepte und Methoden, Verfahren, Techniken und Technologien zur Analyse von massiv großen Datensätzen zu verstehen und zu bewerten. Mögliche Inhalte umfassen:</p> <ul style="list-style-type: none"> • Grundlagen des Information Retrieval • Ähnlichkeitssuche und Clustering • Analyse von Datenströmen und temporalen Daten • Webgraphen: Linkanalyse und soziale Netzwerke • Dynamische Netzwerke und Informationsausbreitung • Empfehlungssysteme und Onlinewerbung • Berechnungsverfahren für massive Datensätze <p>Außerdem können die Studierenden fachliche Lösungskonzepte zur Analyse großer Datensätze in Programme umsetzen.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten; Fachspezifische Vertiefung; Umsetzen fachlicher Lösungskonzepte in Programme und Modelle; Kenntnisse der Vor-/Nachteile von Entwurfsalternativen und Bewertung im jeweiligen Anwendungszusammenhang; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit, mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen anzustellen; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen;</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)</p>		
<p>Voraussetzungen: Modul Datenbanksysteme (INF-0073) - empfohlen Modul Diskrete Strukturen für Informatiker (INF-0109) - empfohlen Modul Informatik 3 (INF-0111) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Analyzing Massive Data Sets (Vorlesung) Lehrformen: Vorlesung Sprache: Englisch SWS: 4</p>		

Inhalte:

Die Vorlesung behandelt grundlegende Konzepte zur Analyse massiv großer Datensätze wie Informationsextraktion, Ähnlichkeitssuche, Clustering, Link- und Netzwerkanalyse sowie deren Implementierung.

Literatur:

- Mining of Massive Datasets. J. Leskovec, A. Rajaraman, J.D. Ullman. Cambridge University Press, 2014
- D. Easley, J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, 2010.
- R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval

Weitere Literatur wird in der Vorlesung bekannt gegeben

Modulteil: Analyzing Massive Data Sets (Übung)

Lehrformen: Übung

Sprache: Englisch / Deutsch

SWS: 2

Prüfung

Analyzing Massive Data Sets

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0284: Praktikum Mobile Application Development		5 ECTS/LP
Version 1.0.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Students learn to realise concepts and models of mobile application development on the Android platform. They learn how to acquire data and signals from different sensors and to implement algorithms of pattern recognition and data/signal analysis on mobile devices. After participation in the Praktikum, students know how to analyse and structure complex problems in the field and to select suitable and state-of-the-art approaches to their solution.</p> <p>Participants are trained in their analytical and conceptional skills as well as in practical programming skills to transfer their knowledge to a practical task. They learn how to make scientifically meaningful assessments of their system using appropriate methods. They will know the mindset and thinking of application development and software engineering. All knowledge obtained during the Praktikum is applied in practice-oriented tasks. Furthermore, after participation, students will be able to recognise important technical evolution in the field of sensors, mobile apps and related tools, and smart devices.</p> <p>Students will work in teams and organise their work and task distribution in an autonomous way. They will learn how to summarise, present and document results in a reasonable way.</p> <p>Key skills: Formal methods; Methods for software development and abstraction; Versioning tools; Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams, Understanding of team management; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality Awareness.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
Voraussetzungen:		
Programming skills in Java are required.		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Praktikum Mobile Application Development</p> <p>Lehrformen: Praktikum</p> <p>Sprache: Englisch</p> <p>SWS: 4</p>
<p>Inhalte:</p> <p>In the Praktikum Mobile Application Development, students will develop software for the Android platform in groups of 2-4 students.</p> <p>Students will get to know and to program several groups of Android devices, e.g., mobiles and wearables, and respective sensors. They will familiarise with the Android Studio IDE, the according toolchain and the code management tool Gi.</p> <p>Topics: Java, Android, App development, UI design, Audio Player, Map and Location, Camera control, Mobile devices, Intelligent signal processing, Wearables.</p>

Literatur:

The literature is going to be announced by the lecturers during the course.

Prüfung

Praktikum Mobile Application Development

Praktikum

Modul INF-0291: Praktikum Selbstlernende Systeme		8 ECTS/LP
Version 1.2.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Praktikum sind die Studierenden in der Lage, Problemstellungen (höherer Komplexität) in dem Bereich „maschinelle Lernverfahren“ zu verstehen und zu lösen. Sie können unterschiedliche Verfahren vergleichen und einordnen und diese eigenständig auf konkrete Beispiele aus der Praxis anwenden. Studenten können intelligente Systeme im Bezug auf die algorithmische Lösung bewerten und sind weiterhin mit Verfahren zur Leistungsevaluierung eines intelligenten Systems vertraut. Sie sind außerdem 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. Weiterhin können sie bedeutende technische Enticklungen im Bereich „maschinelles Lernen“ erkennen und einordnen.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> - Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen, Fertigkeiten zur Entwicklung und Umsetzung von Lösungsstrategien - Kompetenz zur Vernetzung unterschiedlicher Fachgebiete - Präsentation und Dokumentation (eigener) Ergebnisse - Analytische methodische Kompetenz - Fähigkeit produktiv und zielführend im Team zu arbeiten - Akribisches Arbeiten - Fachübergreifende Kenntnisse - Systematische Weiterentwicklung von Entwurfsmodellen - Zeitmanagement - Eigenständige Literaturrecherche zu angrenzenden Themen - Grundsätze guter wissenschaftlicher Praxis 		
<p>Arbeitsaufwand: Gesamt: 240 Std. 225 Std. Praktikum (Selbststudium) 15 Std. Teilnahme an Lehrveranstaltungen (Präsenzstudium)</p>		
<p>Voraussetzungen: Empfohlen: die erfolgreiche Teilnahme an der Vorlesung OC2, Programmiererfahrung, Teamfähigkeit Modul Organic Computing II (INF-0066) - empfohlen</p>		<p>ECTS/LP-Bedingungen: Bestehen der Modulprüfung</p>
<p>Angebotshäufigkeit: jedes Semester</p>	<p>Empfohlenes Fachsemester: ab dem 2.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 2</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Praktikum Selbstlernende Systeme Lehrformen: Praktikum Dozenten: Prof. Dr. Jörg Hähner Sprache: Deutsch SWS: 2 ECTS/LP: 8</p>		

Inhalte:

In dem Praktikum „Selbstlernende Systeme“ sollen die Studenten unterschiedliche Methoden aus dem Bereich „machine learning“ kennenlernen und vor allem selber implementieren. In einem Einführungskurs werden die benötigten theoretischen Grundlagen gelegt und diese dann innerhalb von Kleingruppen an konkreten Beispielen aus der Praxis angewandt.

Literatur:

aktuelle wissenschaftliche Paper

Zugeordnete Lehrveranstaltungen:

Praktikum zu Selbstlernende Systeme (Praktikum)

In dem Praktikum „Selbstlernende Systeme“ sollen die Studenten unterschiedliche Methoden aus dem Bereich „machine learning“ kennenlernen und vor allem selber implementieren. In einem Einführungskurs werden die benötigten theoretischen Grundlagen gelegt und diese dann innerhalb von Kleingruppen an konkreten Beispielen aus der Praxis angewandt.

Prüfung

Praktikum Selbstlernende Systeme

Portfolioprüfung, Kombination aus Praktischer und Schriftlich-Mündlicher Prüfung

Beschreibung:

schriftliche Abgaben, Softwareabnahme, Abschlussvortrag

Modul ZCS-2000: Softskills		2 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Claudia Lange-Hetmann		
<p>Lernziele/Kompetenzen:</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"> - 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. - 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. - 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 oder vertiefen Teilaspekte wie u.a. Kreativität, Innovationsfähigkeit mit innovativen Methoden. <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>Bemerkung:</p> <p><u>Anmeldungspflicht:</u> Für die Teilnahme an den Kursen ist eine Anmeldung über digicampus/ VV erforderlich. <u>Anmeldephase:</u> 10. Jan – 31. Jan (für das folgende SS) bzw. 10. Juli – 31. Juli (für das folgende WS). Die Kurse finden größtenteils ab Mitte März bis letzten Sa* im April (SS) bzw. ab Mitte Sep. bis letzten Sa* im Okt. *vor Vorlesungsbeginn statt und haben eine limitierte Teilnehmerzahl pro Semester.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 60 Std.</p> <p>20 Std. Seminar (Präsenzstudium)</p> <p>5 Std. Vorbereitung von Präsentationen (Selbststudium)</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>Voraussetzungen:</p> <p>keine</p>		<p>ECTS/LP-Bedingungen:</p> <p>aktive Übungsteilnahme im Kurs</p>
<p>Angebotshäufigkeit: jedes Semester</p>	<p>Empfohlenes Fachsemester:</p> <p>ab dem 2.</p>	<p>Minimale Dauer des Moduls:</p> <p>1 Semester</p>

SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
------------------	---	--

<p>Moduleile</p> <p>Modulteil: Softskills</p> <p>Lehrformen: Kurs</p> <p>Sprache: Deutsch / Englisch</p> <p>SWS: 2</p> <p>ECTS/LP: 2</p> <p>Inhalte:</p> <p>Zur Auswahl stehen nachfolgende Kurse/Themen:</p> <p><u>(1) Kommunikationskompetenz</u></p> <ul style="list-style-type: none"> - Kommunikationstraining - Rhetorik - Präsentation - strategische Gesprächsführung <p><u>(2) Sozialkompetenz</u></p> <ul style="list-style-type: none"> - Konfliktmanagement - Moderation & Teamleitung - Führungskompetenzen entwickeln - Führungskompetenzen 4.0 - Agiles Management <p><u>(3) Methodenkompetenz</u></p> <ul style="list-style-type: none"> - Zeit-/Selbst-/Changemanagement - Besprechungsmanagement - Innovationen entwickeln - Design Thinking - Projektmanagement - Projektbasiertes Unternehmertum - Unternehmerisches Denken - U-Planspiel <p>Weiterehin können auch Kompakt-Kurse gewählt werden, bei denen die Teilnehmer o.g. Fähigkeiten erlernen und eine Projektaufgabe im Team bearbeiten:</p> <p>Kompaktkurs Märkte für Menschen</p> <p>Kompaktkurs Entrepreneurship</p> <p>ACE Startup-Challenge</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 http://www.uni-augsburg.de/de/career-service/studierende/veranstaltungen_fakultaet/ bzw. im VV Anmeldesystem https://thi-vv.informatik.uni-augsburg.de/vv/</p>

Literatur:

- (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
- (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
- (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 & Yves Pigneur, 2010. Campus Verlag.

Zugeordnete Lehrveranstaltungen:

Kompaktkurs - Märkte für Menschen (Kurs)

Inhalte: In diesem Kompaktkurs werden Inhalte von gesellschaftlicher Relevanz interdisziplinär erschlossen. Hierbei werden die folgenden drei Themenbereiche kontrovers diskutiert : • Ernährungsgewohnheiten vs. globale Ernährungssicherheit • (Finanz-)Märkte vs. Gerechtigkeit • Leistungsgesellschaft vs. sozialer Systemgedanke Jeder dieser Themenbereiche wird zunächst mit Hilfe wissenschaftlicher Inputs interdisziplinär erschlossen. Daran anschließend erarbeitet sich jede Kleingruppe fundiertes Wissen zu ihrem Themenbereich, setzt sich differenziert mit kontroversen Perspektiven auseinander und präsentiert die Ergebnisse anschließend kontrovers in einer „Talk-Show“ bzw. Podiumsdiskussion. Die Studierenden erarbeiten in Projektgruppen selbst konkrete Vorschläge und Projekte zur Gestaltung einer besseren Zukunft im Rahmen der obigen Themenkomplexe und präsentieren diese zum Abschluss des Seminars. Parallel zu dieser inhaltlichen Arbeit werden methodische Kenntnisse und Hintergrundinf ... (weiter siehe Digicampus)

Soft-Skill-Kurse WS 2018/19 für MPing_I (Kurs)

Startup Challenge - Augsburg Center for Entrepreneurship (Bachelorseminar)

• Innovatives Entrepreneurship Format und Startup Projektstudium • Anhand einer realen unternehmerischen Entscheidungssituation können die Studierenden ihr ganzes unternehmerisches Talent testen • Die Studierenden lernen wesentliche Methoden und Konzepte zur Steuerung und Bewertung von Geschäftsmodellen, Pricing Strategien, Vertrieb und Marketing. • Experten und renommierte Startup-Mentoren begleiten die Studierenden in fachlichen Fragen. • Fachvorträge zu ausgewählten Themenstellungen runden das Seminarangebot ab; sowie die Möglichkeit mit der Augsburger Startup Szene auf Tuchfühlung zu gehen.

Startup Challenge - Augsburg Center for Entrepreneurship (Masterseminar) (Seminar)

• Innovatives Entrepreneurship Format und Startup Projektseminar • Anhand einer realen unternehmerischen Entscheidungssituation können die Studierenden ihr ganzes unternehmerisches Talent testen • Die Studierenden lernen wesentliche Methoden und Konzepte zur Steuerung und Bewertung von Geschäftsmodellen, Pricing Strategien, Vertrieb und Marketing. • Experten und renommierte Startup-Mentoren begleiten die Studierenden in fachlichen Fragen. • Fachvorträge zu ausgewählten Themenstellungen runden das Seminarangebot ab; sowie die Möglichkeit mit der Augsburger Startup Szene auf Tuchfühlung zu gehen.

Prüfung

Anwesenheit und aktive Übungsteilnahme im Kurs

Beteiligungsnachweis, unbenotet

Modul INF-0067: Peer-to-Peer und Cloud Computing		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen: Die erfolgreiche Teilnahme an dieser Lehrveranstaltung resultiert in fachlichen Kompetenzen in zwei Teilbereichen: Peer-to-Peer-Systeme und Cloud-Computing.</p> <p>Peer-to-Peer-Systeme Die Studierenden können verschiedene Peer-to-Peer-System-Architekturen und deren Vor- und Nachteile beschreiben. Insbesondere sind sie in der Lage, für einen gegebenen Anwendungsfall eine passende Architektur auszuwählen und diese Entscheidung zu begründen. Des Weiteren können sie gängige dezentrale Suchalgorithmen und -heuristiken einordnen und bezüglich ihrer Effizienz vergleichen. Sie verstehen abstrakte Netzwerkmodelle, und können mit deren Hilfe Peer-to-Peer-Netzwerke analysieren.</p> <p>Cloud-Computing Teilnehmer können die dem Cloud-Computing zugrunde liegenden Technologien unterscheiden und basierend auf diesem Wissen Cloud-Computing-Architekturen analysieren. Sie kennen die Wichtigkeit von Lastverteilung im Rahmen von großen Cloud-Computing-Anwendungen und können die verschiedenen Ansätze dafür erklären und gegeneinander abwägen. Die Studierenden kennen moderne Ansätze zu nebenläufigen Berechnungen über große Datenmengen und können diese auf einfache Probleme anwenden. Sie sind in der Lage, einfache Anwendungen mithilfe von Platform-as-a-Service-Frameworks zu entwickeln.</p> <p>Schlüsselqualifikationen Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen.</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Peer-to-Peer und Cloud Computing (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		

Inhalte:

Dieses Modul vermittelt Ansätze zur Organisation von komplexen abstrahierten IT-Infrastrukturen, die dynamisch an wechselnde Nutzungsbedingungen angepasst werden können und Dienste wie Rechenkapazität, Datenspeicher, Netzkapazitäten oder Software auf verschiedenen Ebenen zur Verfügung stellen. Dazu werden die Anforderungen, Eigenschaften 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.

Literatur:

- aktuelle wissenschaftliche Paper
- Mahlmann und Schindelhauer: Peer-to-Peer Netzwerke - Algorithmen und Methoden, Springer 2007
- Antonopoulos und Gillam: Cloud Computing - Principles, Systems and Applications, Springer 2010

Zugeordnete Lehrveranstaltungen:

Peer-to-Peer und Cloud Computing (Vorlesung)

Die Vorlesung behandelt u.a. folgende Themen: • Netzwerkmodelle • strukturierte Peer-to-Peer-Systeme – Distributed Hash Tables – Chord – CAN • unstrukturierte Peer-to-Peer-Systeme – Suche – Caching und Replikation – Technologien der verschiedenen Gnutella-Netzwerke • Cloud Computing–Technologien – Virtualisierung – Service-orientierte Architekturen – Cloud-Architekturen – Load Balancing–Ansätze

Modulteil: Peer-to-Peer und Cloud Computing (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Peer-to-Peer und Cloud Computing (Übung)

Die zum Modul „Peer-to-Peer und Cloud Computing“ zugehörige Übung bietet die Möglichkeit, die in der Vorlesung erlernten Konzepte zu vertiefen und beispielhaft anzuwenden. Dazu gehören u.a. die Analyse von wissenschaftlichen Beiträgen zu einigen Schwerpunkten der Vorlesung sowie das Lösen anwendungsorientierter Aufgaben.

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.

Modul INF-0069: Weiterführende Betriebssystemkonzepte		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen: Die Veranstaltung vermittelt den Studierenden bei erfolgreicher Teilnahme fundierte fachliche Kenntnisse über moderne Betriebssysteme und die dahinterstehenden grundlegenden Konzepte. Sie können danach die Geschichte der Betriebssysteme wiedergeben und deren primäre Typen erläutern. Der theoretische Teil konzentriert sich auf die Vertiefung des Verständnisses von Mechanismen aus den Bereichen Multiprogramming, Interprozesskommunikation, Speicher-Management und Virtualisierung. Die Studierenden können die kennengelernten Ansätze hinsichtlich ihrer Eignung für einen bestimmten Einsatzzweck beurteilen. Der praktische Teil befasst sich mit dem Transfer und der Anwendung des im theoretischen Teil vermittelten Wissens. Sie benutzen Programme und Programmierschnittstellen aus dem Unix-Umfeld, um ein erarbeitetes Lösungskonzept mit dem Schwerpunkt Ein-/Ausgabe durch einen Programmcode zu realisieren.</p> <p>Schlüsselqualifikationen: Qualitätsbewusstsein, Akribie; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Kenntnisse der Vor-/Nachteile von Entwurfsalternativen, Bewertung im jeweiligen Anwendungszusammenhang; Fachspezifische Vertiefungen; Umsetzen fachlicher Lösungskonzepte in Programme und Modelle.</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 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) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Weiterführende Betriebssystemkonzepte (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		

Inhalte:

Die Vorlesung beinhaltet u. a. folgende Themen:

- Grundlagen eines Betriebssystems
 - Aufgaben
 - Struktur und Aufbau
 - Historie
 - Shell
- Prozesse, Threads und Interrupts
 - Prozesse und Threads
 - HW-Interrupts
- Scheduling
 - Grundlagen des Scheduling
 - Echtzeit-Scheduling
- Speicher
 - Interaktion
 - Speicherkonzepte
 - Dateisysteme
- Virtualisierung

Literatur:

- Folien
- Andrew S. Tanenbaum: "Moderne Betriebssysteme", Pearson Studium, ISBN:978-3-8273-7342-7

Modulteil: Weiterführende Betriebssystemkonzepte (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Weiterführende Betriebssystemkonzepte

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

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

Modul INF-0145: Mikrorechnertechnik und Echtzeitsysteme		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen in den folgenden Bereichen auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: Grundlagen und Aufbau von Mikrorechnern und Mikrocontrollern, Echtzeitprogrammierung und -betriebssysteme, funktionale Sicherheit, Automotive- und Avionic-Systeme. Sie verstehen die Konzepte hinter und den Aufbau von Mikrocontrollern und können einordnen, wo diese eingesetzt werden. Sie lernen Kriterien, anhand derer sie auswählen können, welcher Mikrocontroller für welchen Zweck sinnvoll eingesetzt werden kann. Dazu können die Studierenden beispielsweise deren Energieeffizienz einschätzen. Um die Erkenntnisse direkt in die Praxis zu übertragen, analysieren die Studierenden am Beispiel einer Wetterstation, wie ein konkretes System realisiert werden kann.</p> <p>Sie können die verschiedenen Arten von Echtzeitsystemen differenzieren und entscheiden, welche Echtzeit-Schedules für ein bestimmtes Echtzeitsystem geeignet sind. Weiterhin lernen die Studierenden verschiedene Techniken der Echtzeitprogrammierung, was auch die Auswahl der zu einer Anwendung passenden Ablaufsteuerung miteinschließt. Schließlich werden noch Standards der funktionalen Sicherheit (SIL, ASIL) bewertet.</p> <p>Schlüsselqualifikationen: Analytisch-methodische Kompetenz, Abwägung von Lösungsansätzen, Präsentation von Lösungen von Übungsaufgaben; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein, Akribie; Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 15 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium) 75 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Mikrorechnertechnik und Echtzeitsysteme (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 3</p>		

Inhalte:

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.

Literatur:

- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010
- Heinz Wörn, Uwe Brinkschulte, Echtzeitsysteme, Springer Verlag, Berlin/Heidelberg, 2005

Zugeordnete Lehrveranstaltungen:**Mikrorechnertechnik und Echtzeitsysteme (Vorlesung)**

Die MA-Vorlesung "Mikrorechnertechnik und Echtzeitsysteme" behandelt aufbauend auf der BA-Vorlesung "Systemnahe Informatik" die grundlegenden Prinzipien der Mikrocontroller und der eingebetteten Systeme. In der Praxis häufig verwendete Mikrocontroller und Bustechnologien werden in ihrer Funktionsweise analysiert und zukunftsweisende Technologien dieser Bausteine erläutert. Weiterhin wird auf Anforderungen eingebetteter Echtzeitsysteme eingegangen. Echtzeitanwendungen finden sich im Flugzeugbau, in der Motorsteuerung und in Fahrerassistenzsystemen in Autos, in der Kraftwerkssteuerung und in vielen industriellen Maschinen. Für solche Anwendungen werden die Grundlagen der Echtzeitsysteme bereitgestellt.

Modulteil: Mikrorechnertechnik und Echtzeitsysteme (Übung)**Lehrformen:** Übung**Sprache:** Deutsch**SWS:** 1**Zugeordnete Lehrveranstaltungen:****Übung zu Mikrorechnertechnik und Echtzeitsysteme (Übung)****Prüfung****Mikrorechnertechnik und Echtzeitsysteme (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0147: Prozessorarchitektur		5 ECTS/LP
Version 1.1.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden verstehen die Prinzipien des Aufbaus folgender Prozessoren auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: superskalare Mikroprozessoren, Multi- und Manycore-Prozessoren, Signalprozessoren, Smartphone-Prozessoren und GPUs. Außerdem lernen sie aktuelle Konzepte der Prozessorarchitektur kennen und können Vor-/Nachteile aktueller und zukünftiger Prozessoren anhand ihres internen Aufbaus einschätzen und beurteilen. Sie verstehen die Funktionsweise von verschiedenen Komponenten von Mikroprozessoren und deren Zusammenspiel. Somit ist es den Studierenden möglich, den Einfluss verschiedener Architekturerweiterungen auf das Gesamtsystem einzuordnen. Ebenso können sie unterscheiden, wann es sinnvoll ist, speichergekoppelte, nachrichtengekoppelte oder datenparallele Prozessoren einzusetzen, und differenzieren zwischen Techniken für energieeffiziente oder hochperformante Prozessoren. In der Übung simulieren die Studierenden verschiedene Prozessorarchitekturen und Sprungvorhersagetechniken und bewerten deren Einfluss auf die Laufzeit und den Energieverbrauch.</p> <p>Schlüsselqualifikationen: Analytisch-methodische Kompetenz, Abwägung von Lösungsansätzen, Präsentation von Lösungen von Übungsaufgaben; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein, Akribie; Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen; Fertigkeit, mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen anzustellen; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen		
Angebotshäufigkeit: unregelmäßig (i. d. R. im WS)	Empfohlenes Fachsemester: 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Prozessorarchitektur (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: Die Vorlesung "Prozessorarchitektur" vertieft die Techniken superskalarer Mikroprozessoren und aktueller Multicore-Prozessoren. Dabei werden die Pipelinestufen detailliert behandelt, mehrfädige Prozessoren und Multicores gegenübergestellt sowie aktuelle Beispielprozessoren vorgestellt. Außerdem wird aus der Forschung an Manycores und Echtzeit-Multicores berichtet.</p>		

Literatur:

- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010
- John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011

Zugeordnete Lehrveranstaltungen:

Prozessorarchitektur (Vorlesung)

Ausgehend von den Herausforderungen bei der Leistungssteigerung moderner Prozessoren werden in der Vorlesung bereits realisierte Konzepte wie Superskalartechnik, Mehrfädigkeit und Out-of-Order-Ausführung behandelt. Darauf aufbauend werden weitere Architekturtypen wie GPUs vorgestellt sowie aktuelle Forschungsthemen, z.B. Manycore-Prozessoren, betrachtet.

Modulteil: Prozessorarchitektur (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Prozessorarchitektur (Übung)

Prüfung

Prozessorarchitektur (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

Modul INF-0148: Entwurf und Analyse fehlertolerierender Rechensysteme		5 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden verstehen die folgenden Konzepte auf einem fortgeschrittenen, wissenschaftlichen Niveau: Ursachen, Effekte und Modellierung von Hardware-Fehlern, stochastische Grundlagen der Fehlerrechnung, Modellierung und Analyse von Rechensystemen mit Zuverlässigkeitsblockdiagrammen, Fehlerbäumen und FMEA, unterschiedliche Redundanzarten und ihre Verwendung in verschiedenen Hard- und Software-Techniken zur Fehlererkennung, Fehlerkorrektur, Fehlerdiagnose und Rekonfiguration. Weitere Schwerpunkte sind fehlertolerierende Speicherstrukturen und Fehlertoleranzkonzepte in sicherheitskritischen Systemen am Beispiel der ISO 26262 Norm.</p> <p>Die Studierenden können die Ursachen und Effekte von Hardware-Fehlern nachvollziehen und sind in der Lage unterschiedliche Redundanztechniken analysieren und bewerten zu können. Sie lernen verschiedene Konzepte fehlertolerierender Rechensysteme aus Theorie und Praxis, wie z.B. Lockstep-Ausführung, Prozess-Redundanz, ECC-Speicher-Implementierungen oder Kontrollflussüberprüfung, kennen, und können deren jeweilige Vor- und Nachteile einschätzen. In den praktischen Übungen wenden die Studierenden die Methoden der Modellierung und Fehlerrechnung an verschiedenen Aufgaben an, präsentieren ihre Lösungen und vergleichen auf qualitativer Ebene verschiedene Fehlertoleranzkonzepte aus Forschung und Praxis.</p> <p>Schlüsselqualifikationen: Analytisch-methodische Kompetenz in der Analyse fehlertolerierender Rechensysteme, eigenständiges Arbeiten mit wissenschaftlichen Aufsätzen, Abwägung von Lösungsansätzen, Verständliche Präsentation von Übungsaufgaben, Selbstreflexion</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 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) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Entwurf und Analyse fehlertolerierender Rechensysteme (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		

Inhalte:

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.

Literatur:

- D.P. Siewiorek, R.S. Swarz: Reliable Computer Systems, Peters, 1998
- I. Koren, C.M. Krishna: Fault Tolerant Systems, Morgan Kaufmann, 2007
- T. Anderson, P.A. Lee: Fault Tolerance - Principles and Practice, Prentice Hall, 1982

Modulteil: Entwurf und Analyse fehlertolerierender Rechensysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Entwurf und Analyse fehlertolerierender Rechensysteme (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

Modul INF-0149: Praktikum Eingebettete Systeme		5 ECTS/LP
Version 1.1.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen in den folgenden Bereichen auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: selbstständiges Arbeiten mit Mikrocontrollern, Datenblättern und Spezifikationen, Anbindung von analoger und digitaler Peripherie, Entwurf und Modellierung von eingebetteter Software mit Zustandsdiagrammen sowie deren Transformation in Code. Weitere Schwerpunkte sind die Konfiguration von sequentiellen Schnittstellen sowie Scheduling und task-basierte Programmierung. Durch die Implementierung von Aufgaben für einen Mikrocontroller setzen sie die erlernten Konzepte direkt in die Praxis um. Dabei steht Interaktion mit Sensoren und Aktoren sowie die Kommunikation mit anderen Systemteilen im Vordergrund. Dazu können sie unterschiedliche Arten der Ablaufsteuerung unterscheiden und anwenden. In einer abschließenden Projektphase lernen die Studierenden, eine komplexe Aufgabenstellung im Team zu planen, nach einem selbst entwickelten fundierten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zur verständlichen Darstellung und Dokumentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein, Akribie; Projektgebundene Arbeit und Zeitmanagement; Auswahl und sichere Anwendung geeigneter Methoden; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Selbstreflexion</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p>Voraussetzungen: Kenntnisse in C. Modul Mikrorechner-technik und Echtzeitsysteme (INF-0145) - empfohlen</p>		
<p>Angebotshäufigkeit: jedes Sommersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 4</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Praktikum Eingebettete Systeme Lehrformen: Praktikum Sprache: Deutsch SWS: 4</p>		
<p>Inhalte: 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.</p>		

Literatur:

- Marwedel, Wehmeyer: Eingebettete Systeme, Springer Verlag, Heidelberg, 2007
- Wiegmann: Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller, 6. Auflage, VDE Verlag, Berlin, 2011

Prüfung

Erfolgreiche Praktikumsteilnahme, Projektvorstellung

Praktikum

Modul INF-0150: Hardware-Entwurf		8 ECTS/LP
Version 1.1.0 (seit WS14/15) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen in den folgenden Bereichen auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: Entwurfsprozess für digitale Schaltungen, Schaltungslogik und Gatter, physikalische Grundlagen der elektronischen Bauteile, Beschreibung von Hardware mit einer Hardwarebeschreibungssprache. Zunächst lernen die Studierenden die Verknüpfung logischer Gatter und bauen einen Halb- und einen Volladdierer. Sie verstehen den Entwicklungsprozess für digitale Schaltungen und wenden diesen direkt praktisch an, indem sie ihren eigenen RISC-V-Prozessor entwickeln. Sie modellieren und implementieren ihn selbstständig mit Hilfe der Hardwarebeschreibungssprache VHDL. Dazu lernen sie die Vor- und Nachteile von schematischer und textueller Hardware-Beschreibung kennen und können entscheiden, wann es sinnvoll ist, welche Variante einzusetzen. Darüber hinaus kombinieren sie synchrone und asynchrone Prozesse, um ein gutes Zusammenspiel der Komponenten ihres selbstgebauten Mikroprozessors zu erreichen. Schließlich beurteilen die Studierenden die Effizienz ihrer Implementierung anhand der erzielten Taktfrequenz und des benötigten Hardware-Aufwandes. In einer abschließenden Projektphase lernen sie, eine komplexe Aufgabenstellung zu planen, nach einem selbst entwickelten fundierten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zur verständlichen Darstellung und Dokumentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein, Akribie; Projektgebundene Arbeit und Zeitmanagement; Auswahl und sichere Anwendung geeigneter Methoden; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Selbstreflexion</p>		
<p>Bemerkung: Im WiSe 2018/2019 findet statt diesem Modul das Modul "INF-0297: Praktikum Prozessorbau" statt. Das weitere Angebot dieses Moduls ist derzeit noch nicht abzusehen.</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium)</p>		
<p>Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Hardware-Entwurf (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: 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</p>		

Literatur:

- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010
- John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011

Modulteil: Hardware-Entwurf (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Projektvorstellung und Projektabnahme

Praktikum

Beschreibung:

Erfolgreiche Praktikumsteilnahme, Projektvorstellung am Ende des Semesters

Modul INF-0216: Vertiefte Multicore-Programmierung		8 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen auf einem fortgeschrittenen, wissenschaftlichen Niveau in folgenden Bereichen: Parallelprogrammierung von speichergekoppelten Systemen in C++11 und OpenMP, Entwurf von lock-freien Algorithmen, Programmierung eines Hardware-Transaktionsspeicher-Systems am Beispiel von Intel TSX, nachrichtengekoppelte Programmierung mit dem Message-Passing-Interface (MPI), Programmierung von GPU-Beschleunigerkarten mit Nvidia CUDA.</p> <p>Sie können die Vor- und Nachteile der unterschiedlichen Programmiermodelle und Architekturen einschätzen, und parallele Programme in den jeweiligen Sprachen analysieren. Durch die praktischen Übungen setzen die die Studierenden die erlernten Konzepte direkt in die Praxis um. Dabei erwerben die Studierenden fortgeschrittene Programmierkenntnisse in C++11, Intel TSX, MPI, OpenMP und CUDA. Sie können selbständig das Laufzeitverhalten der jeweiligen Programmiermodelle bewerten und optimieren. In einer abschließenden Projektphase lernen die Studierenden, eine komplexe Aufgabenstellung im Team zu planen, nach einem selbst entwickelten Lösungsansatz zu bearbeiten und die Resultate in wissenschaftlicher Weise im Plenum zu diskutieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Analytische-methodische Kompetenz im Bereich der Multicore-Programmierung von CPUs und GPUs; Projektgebundene Arbeit und Zeitmanagement; Auswahl und sichere Anwendung geeigneter Methoden; Fähigkeit, vorhandenes Wissen selbständig zu kombinieren und zu erweitern; Selbstreflexion</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 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) 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Moduleile		
Modulteil: Vertiefte Multicore-Programmierung (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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.</p>		

Literatur:

- M. Herlihy, N. Shavit: The Art of Multiprocessor-Programming, Morgan Kaufmann, 2012, ISBN: 978-0123973375
- M. McCool, J. Reinders, A. D. Robison: Structured Parallel Programming: Patterns for Efficient Computation, Morgan Kaufmann, 2012, ISBN: 978-0124159938
- T. Rauber, G. Runger: Parallele Programmierung, Springer Verlag, 2007, ISBN: 978-3540465492
- es werden die jeweils neuesten APIs/Unterlagen aus dem Internet verwendet

Modulteil: Vertiefte Multicore-Programmierung (ubung)

Lehrformen: ubung

Sprache: Deutsch

SWS: 4

Inhalte:

Ziel der ubung ist es, den Umgang mit den unterschiedlichen Programmiermodellen sowie Performanzanalyse- und Debugging-Techniken in praktischen Beispielen zu vertiefen. Die ubung wird durch eine Projektphase abgeschlossen, die es den Studierenden ermoglicht, die behandelten Programmier-techniken in einem umfangreicheren Projekt selbstandig anzuwenden, Ergebnisse auszuwerten und zu prasentieren.

Prufung

Vertiefte Multicore-Programmierung (mundliche Prufung)

Mundliche Prufung, Dauer: 30-45 Minuten

Beschreibung:

Projektvorstellung und -abnahme, Fragen zu Vorlesung und ubung

Modul INF-0297: Praktikum Prozessorbau		5 ECTS/LP
Version 1.1.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen in den folgenden Bereichen auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: Entwurfsprozess für digitale Schaltungen, Schaltungslogik und Gatter, physikalische Grundlagen der elektronischen Bauteile, Beschreibung von Hardware mit einer Hardwarebeschreibungssprache. Zunächst lernen die Studierenden die Verknüpfung logischer Gatter und bauen einen Halb- und einen Volladdierer. Sie verstehen den Entwicklungsprozess für digitale Schaltungen und wenden diesen direkt praktisch an, indem sie ihren eigenen RISC-V-Prozessor entwickeln. Sie modellieren und implementieren ihn selbstständig mit Hilfe der Hardwarebeschreibungssprache VHDL. Dazu lernen sie die Vor- und Nachteile von schematischer und textueller Hardware-Beschreibung kennen und können entscheiden, wann es sinnvoll ist, welche Variante einzusetzen. Darüber hinaus kombinieren sie synchrone und asynchrone Prozesse, um ein gutes Zusammenspiel der Komponenten ihres selbstgebauten Mikroprozessors zu erreichen. Schließlich beurteilen die Studierenden die Effizienz ihrer Implementierung anhand der erzielten Taktfrequenz und des benötigten Hardware-Aufwandes. In einer abschließenden Projektphase lernen sie, eine komplexe Aufgabenstellung zu planen, nach einem selbst entwickelten fundierten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zur verständlichen Darstellung und Dokumentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein, Akribie; Projektgebundene Arbeit und Zeitmanagement; Auswahl und sichere Anwendung geeigneter Methoden; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Selbstreflexion</p>		
<p>Bemerkung: Im WiSe 2018/2019 statt "INF-0150: Hardware-Entwurf"</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p>Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen Modul Prozessorarchitektur (INF-0147) - empfohlen</p>		
Angebotshäufigkeit: einmalig WS	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Praktikum Prozessorbau		
Lehrformen: Praktikum		
Sprache: Deutsch		
SWS: 4		
<p>Inhalte: Die Veranstaltung "Prozessorbau" 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 Prozessorphipeline veranschaulicht. Das Ergebnis ist ein in VHDL entwickelter lauffähiger Prozessor für ein FPGA-Prototypenboard</p>		

Literatur:

- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010
- John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011

Zugeordnete Lehrveranstaltungen:

Praktikum Prozessorbau (Praktikum)

Prüfung

Praktikum Prozessorbau

praktische Prüfung

Modul INF-0088: Bayesian Networks		5 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: Bayessche Netze gehören zu den vielseitigsten statistischen Methoden des maschinellen Lernens. Nach einer erfolgreichen Teilnahme an diesem Modul verstehen und vertiefen Teilnehmer die Kernprinzipien von Bayesschen Netzen und können diese auf viele praxisrelevante Probleme in unterschiedlichen Fachgebieten anwenden. Diese umfassen unter anderem die Robotik, Websuche, intelligente Agenten, automatisierte Diagnosesysteme und medizinische Systeme. Die Studierenden sind in der Lage, Bayessche Netze zu verstehen, anzuwenden und fachübergreifende Problemstellungen in diesem Kontext zu analysieren und zu evaluieren. Die Teilnahme an diesem Modul fördert Fertigkeiten zum logischen, analytischen und konzeptionellen Denken. Studierende sind in der Lage, mittels Bayesscher Netze wissenschaftlich aussagekräftige Bewertungen zu erstellen.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Logik; umsetzen fachlicher Lösungskonzepte in Modelle; fachübergreifende Kenntnisse; Entwicklung und Umsetzung von Lösungsstrategien komplexer Problemstellungen; systematische Weiterentwicklung von Entwurfsmethoden; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen</p>		
<p>Bemerkung: Die gleichzeitige Einbringung von diesem Modul und INF-0263 ist nicht möglich.</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 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) 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)</p>		
<p>Voraussetzungen: keine</p>		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 5.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Bayesian Networks (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte:</p> <ol style="list-style-type: none"> 1. Basics of Probability Theory 2. Example: Bayesian Network based Face Detection 3. Inference 4. Influence Diagrams 5. Parameter Learning 6. Example: probabilistic Latent Semantic Analysis (pLSA) 		

Literatur:

- Richard E. Neapolitan. Learning Bayesian Networks. Prentice Hall Series in Artificial Intelligence, 2004. ISBN 0-13-012534-2
- Daphne Koller, Nir Friedman. Probabilistic Graphical Models: Principles and Techniques. The MIT Press, 2009. 978-0262013192

Modulteil: Bayesian Networks (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Bayesian Networks (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

Modul INF-0092: Multimedia II: Machine Learning and Computer Vision		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul besitzen die Studierenden fortgeschrittene Kenntnisse des maschinellen Lernens (Entscheidungsbäume, neuronale Netze und tiefe neuronale Netze, Hypothesenevaluation, Instanz-basiertes Lernen, Bayessches Lernen, Lerntheorie), der Datenreduktion (z.B. Hauptkomponentenanalyse), der fortgeschrittenen Bildverarbeitung und des maschinellen Sehens und können diese anwenden. Sie sind in der Lage, wissenschaftlich komplexe Verfahren auf dem Gebiet der multimedialen Datenverarbeitung zu analysieren, zu verstehen und programmatisch umzusetzen, sowie die erlernten Prinzipien auf neue Fragestellungen geeignet anzuwenden. Sie entwickeln Fertigkeiten zur logischen, analytischen und konzeptionellen Denken im Bereich der digitalen Signalverarbeitung und multimedialen Datenverarbeitung.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Logik; umsetzen fachlicher Lösungskonzepte in Modelle; fachübergreifende Kenntnisse; Entwicklung und Umsetzung von Lösungsstrategien komplexer Problemstellungen; systematische Weiterentwicklung von Entwurfsmethoden; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 60 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 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)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Multimedia II: Machine Learning and Computer Vision (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4</p>
<p>Inhalte: Die Vorlesung gibt einen guten Überblick über alle Aspekte des maschinellen Lernens und der maschinellen Extraktion von Informationen aus Multimediadaten (z.B. "GoogleImage 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)</p>

Literatur:

Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.

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

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Multimedia II: Machine Learning and Computer Vision (Klausur)

Klausur / Prüfungsdauer: 120 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

Modul INF-0093: Probabilistic Robotics		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: Nach einer erfolgreichen Teilnahme an diesem Modul verstehen Teilnehmer Grundlagen und vertiefende Fragestellungen und Algorithmen der Robotik (z.B. rekursive Zustandsschätzung, gaußsche- und nicht-parametrische Filter, Kalman Filter, Bewegung und Lokalisierung, Perzeption, Kartierung, SLAM) aus wahrscheinlichkeitstheoretischer Sicht und können erlernte Konzepte auf komplexe, praxisrelevante Aufgabenstellung anwenden. Die Studierenden sind in der Lage, Problemstellungen in diesem Kontext zu analysieren und zu evaluieren. Die Teilnahme an diesem Modul fördert Fertigkeiten zum logischen, analytischen und konzeptionellen Denken auf dem Gebiet der wahrscheinlichkeitstheoretischen Robotik. Studierende können aus den erlernten Konzepten zielgerichtet geeignete Methoden auswählen, sicher anwenden und auf neue, auch fachfremde, Fragestellungen übertragen. Das Modul vermittelt Kompetenzen zum Erkennen aktueller Forschung und bedeutenden technischen Entwicklungen auf diesem Gebiet.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Logik; umsetzen fachlicher Lösungskonzepte in Modelle; fachübergreifende Kenntnisse; Entwicklung und Umsetzung von Lösungsstrategien komplexer Problemstellungen; systematische Weiterentwicklung von Entwurfsmethoden; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	

Moduleile
<p>Moduleil: Probabilistic Robotics (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>
<p>Inhalte:</p> <ol style="list-style-type: none"> 1. Introduction to Probabilistic Robotics 2. Recursive State Estimation 3. Gaussian Filters 4. Nonparametric Filters 5. Robot Motion 6. Robot Perception 7. Mobile Robot Localization: Markow and Gaussian 8. Mobile Robot Localization: Grid and MonteCarlo 9. Occupancy Grid Mapping 10. SLAM

Literatur:

Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic Robotics. Springer Verlag.

Zugeordnete Lehrveranstaltungen:

Probabilistic Robotics (Vorlesung)

In the course of this lecture students will learn how robots can estimate their state (e.g. their pose) in a probabilistic fashion, i.e. in the face of uncertainty. The main focus of this lecture is on the Bayes Filter algorithm which enables robots to estimate their new state after executing a control and to incorporate sensor measurements to update their belief. Various flavors of the Bayes Filter such as the Kalman Filter and the Particle Filter will be discussed in detail. Furthermore, students will get to know different ways to model robot motion and measurements of various types of sensors. The final chapters of the lecture will be on approaches to robot localization, i.e. the problem of the robot having to determine its position on a given map of the environment. Also, the localization problem will be discussed for situations when the robot has to generate a map itself by occupancy grid mapping or simultaneous localization and mapping (SLAM) algorithms.

... (weiter siehe Digicampus)

Modulteil: Probabilistic Robotics (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Probabilistic Robotics (Übung)

Prüfung

Probabilistic Robotics (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

Modul INF-0112: Graphikprogrammierung		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
Lernziele/Kompetenzen: Die Studierenden beherrschen die 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. Schlüsselqualifikationen: 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		
Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Übung (Präsenzstudium) 60 Std. Vorlesung (Präsenzstudium) 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)		
Voraussetzungen: Mathematik für Informatiker I + II (alternativ Analysis I + Lineare Algebra I) empfohlen Modul Informatik 1 (INF-0097) - empfohlen Modul Informatik 2 (INF-0098) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 4.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
Modulteil: Graphikprogrammierung (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4
Inhalte: Koordinaten und Transformationen, Projektionen und Kameramodelle, Sichtbarkeit, Farbmodelle, Beleuchtung und Schattierung, Texturen, Schattenberechnung, Raytracing, Animationstechniken, OpenGL/JOGL
Literatur: <ul style="list-style-type: none"> • Eigenes Skriptum • M. Bender, M. Brill, Computergrafik - ein anwendungsorientiertes Lehrbuch, Hanser 2006 • F. Hill, S. Kelley: Computer graphics using OpenGL, Pearson 2007
Modulteil: Graphikprogrammierung (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 2

Prüfung

Graphikprogrammierung (Klausur)

Klausur / Prüfungsdauer: 120 Minuten

Modul INF-0175: Multimedia I: Usability Engineering		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an der Lehrveranstaltung haben die Studierenden vertiefte Kenntnisse über aktuelle Paradigmen der (Multimodalen) Mensch-Technik Interaktion und Methoden der Gestaltung und Evaluation von neuartigen multimedialen Interaktionstechniken. Sie sind in der Lage im Team Multimediaprojekte durchzuführen und Ihre Lösungsstrategien kritisch zu evaluieren. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen, Leistung von Beiträgen zur Wissenschaft sowie Kompetenzen in der Evaluation von interaktiven Systemen hinsichtlich traditionellen und modernen Nutzeranforderungen. Außerdem verstehen die Studierenden praxisrelevante Fragestellungen.</p> <p>Schlüsselqualifikationen: Fertigkeit der Zusammenarbeit in Teams; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, Beiträge zur Wissenschaft zu leisten; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen; Qualitätsbewusstsein, Akribie.</p>		
<p>Arbeitsaufwand: 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>Voraussetzungen: Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia Grundlagen II (INF-0166) - empfohlen</p>		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Multimedia I: Usability Engineering (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4</p>
<p>Inhalte: Methoden, Werkzeuge und Vorgehensweisen zur Gestaltung von gebrauchstauglichen Softwareprodukten</p>
<p>Literatur:</p> <ul style="list-style-type: none"> • Ben Shneiderman, "Designing the User Interface: Strategies for Effective Human-Computer Interaction" • Jakob Nielsen, "Usability Engineering" • Helen Sharp, Yvonne Rogers und Jenny Preece, "Interaction Design beyond Human Computer Interaction"
<p>Zugeordnete Lehrveranstaltungen: Multimedia 1: Usability Engineering (Vorlesung) Durch die zunehmende Verbreitung von Computern in allen Lebensbereichen und deren Einbettung in die natürliche Umgebung des Benutzers wird die Gestaltung der Mensch-Technik-Interaktion zu einer großen Herausforderung. Während bislang rein technische Aspekte dominierten, geht der Trend zu Entwurfsprozessen, die den Nutzer in den Vordergrund stellen und ihn bereits in der Entwurfsphase miteinbeziehen. Ziel dieser</p>

Veranstaltung ist die praxisnahe Vermittlung von Wissen zum Thema Usability Engineering. Dabei werden unter anderem unterschiedliche Verfahren zur nutzerzentrierten Entwicklung von klassischen und neuartigen Nutzerschnittstellen vorgestellt. Der praktische Teil der Vorlesung startet mit der Entwicklung eines Konzeptes und endet mit der Implementierung eines ersten klickbaren Prototypen.

Modulteil: Multimedia I: Usability Engineering (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Multimedia 1: Usability Engineering (Übung)

Prüfung

schriftliche Abgaben

Übung + Praktikum

Modul INF-0178: Praktikum Usability Engineering		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Die Studierenden sind mit Methoden und Techniken des Usability Engineering vertraut. Nach der erfolgreichen Teilnahme verfügen sie über die notwendigen Kenntnisse, um nach den Vorgaben des Nutzerzentrierten Designprozesses Anwendungsszenarien zu analysieren und zielgruppengerechte Softwarelösungen zu entwerfen. Sie sind dazu fähig, aktuelle Interaktionsparadigmen und Design-Richtlinien in Modelle und Programme für neuartige Interaktionsgeräte zu übersetzen, sowie sich selbstständig in die notwendigen Technologien einzuarbeiten. Des weiteren können sie praxisrelevante Evaluationsmethoden anwenden, um die Qualität des erstellten Softwareprototypen zu bewerten. 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>Schlüsselqualifikationen: 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>Arbeitsaufwand: Gesamt: 240 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 90 Std. Praktikum (Präsenzstudium)</p>		
Voraussetzungen: Modul Multimedia I: Usability Engineering (INF-0175) - empfohlen		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Praktikum Usability Engineering Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		
Inhalte: Die konkrete Aufgabenstellung für Studentenprojekte wird jedes Jahr neu entworfen.		
Literatur: Literaturhinweise werden je nach Thema zu Beginn des Semesters bekanntgegeben.		
<p>Prüfung Vortrag mit Softwarepräsentation, Ausarbeitung mit Softwaredokumentation Projektarbeit</p>		

Modul INF-0179: Einführung in die Spieleprogrammierung		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul verstehen die Studierenden die wesentlichen Konzepte zur Entwicklung eines Spiels. Sie sind in der Lage, eigene Gameplay-Konzepte umzusetzen und passende KI-Verfahren zu Entscheidungsfindung sowie Ablaufsteuerung einzusetzen. Sie verfügen über die Kenntnisse der Denkweise und die Sprache anwendungsrelevanter Disziplinen. Im Rahmen der Vorlesung erlernen sie, KI-Komponenten mit geeigneten Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmik selbstständig zu entwickeln und technisch umzusetzen. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen sowie zum logischen, analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikationen: Auswahl und sichere Anwendung geeigneter Methoden, Fachübergreifende Kenntnisse, Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen, Fertigkeit zur Entwicklung und Umsetzung von Lösungsstrategien für komplexe Probleme, Verstehen von Teamprozessen, Fertigkeit der Zusammenarbeit in Teams, Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen, Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 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) 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)</p>		
<p>Voraussetzungen: Ferienaufgabe</p>		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Einführung in die Spieleprogrammierung (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: Game Engines, Entscheidungsfindung für KI-Charaktere, Wegfindung und Navigation, Gruppenverhalten und Gruppendynamik, Animationen, Physik, Storytelling, Ein-/Ausgabemethodik, Audio, Game Design, Grafik und Shaderprogrammierung</p>		
<p>Literatur: Skript</p>		
<p>Modulteil: Einführung in die Spieleprogrammierung (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 4</p>		

Prüfung

Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation, Übungsaufgaben

Projektarbeit

Modul INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul sind die Studierenden mit den Grundkonzepten der Signalverarbeitung und des maschinellen Lernens vertraut. Sie sind in der Lage, fachliche Lösungskonzepte in Programme und Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Im Rahmen des Praktikums erlernen sie außerdem, 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. Besonders gefördert wird zudem die Fähigkeit geeignete Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmik selbstständig zu entwickeln und technisch umzusetzen.</p> <p>Schlüsselqualifikationen: Fortgeschrittene Kenntnisse der Signalverarbeitung und des maschinellen Lernens, Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen, Fertigkeit zur Entwicklung und Umsetzung von Lösungsstrategien für komplexe Probleme, Verstehen von Teamprozessen, Fertigkeit der Zusammenarbeit in Teams, Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen, Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p>Voraussetzungen: Programmiererfahrung Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia Grundlagen II (INF-0166) - empfohlen</p>		
<p>Angebotshäufigkeit: jedes Sommersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Praktikum Multimodale Echtzeitsignalverarbeitung Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		
<p>Inhalte: Die konkrete Aufgabenstellung aus dem weiten Gebiet der multimodalen Echtzeitsignalverarbeitung wird jedes Jahr neu entworfen.</p>		
<p>Literatur: Literaturhinweise werden je nach Thema zu Beginn des Semesters bekanntgegeben.</p>		
<p>Prüfung Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation Projektarbeit</p>		

Modul INF-0183: Praktikum Spieleprogrammierung		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul verstehen die Studierenden die wesentlichen Konzepte der Spieleentwicklung. 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 zu präsentieren. Sie sind in der Lage, fachliche Lösungskonzepte in Programme und Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Sie verfügen über die Kenntnisse der Denkweise und die Sprache anwendungsrelevanter Disziplinen. Im Rahmen des Praktikums erlernen sie, verschiedenste Komponenten eines Spiels mit geeigneten Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmik selbstständig zu entwickeln und technisch umzusetzen. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen sowie zum logischen, analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikationen: 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>Arbeitsaufwand: Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
Voraussetzungen: Modul Einführung in die Spieleprogrammierung (INF-0179) - empfohlen		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 2.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Praktikum Spieleprogrammierung Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		
<p>Inhalte: Innerhalb des Praktikums soll ein Spiel entwickelt werden. Der inhaltliche Schwerpunkt des Praktikums wird jedes Jahr neu festgelegt.</p>		
<p>Literatur: Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.</p>		
<p>Zugeordnete Lehrveranstaltungen: Praktikum Spieleprogrammierung (Praktikum) Ziel der Veranstaltung ist das Wissen aus der Vorlesung "Einführung in die Spieleprogrammierung" praktisch anzuwenden und zu vertiefen. Voraussetzung ist deshalb die erfolgreiche Teilnahme an dieser Veranstaltung! Im Verlauf des Praktikums werden die Teilnehmer gemeinsam ein Spiel entwickeln. Das genaue Thema des Spiels wird später festgelegt. Die Teilnehmer des Praktikums werden in kleinere Gruppen aufgeteilt, die an</p>		

unterschiedlichen Aufgabenbereichen arbeiten, wie der Visualisierung des Spielgeschehens, der Interaktion des Nutzers mit der Anwendung oder der künstlichen Intelligenz der virtuellen Charaktere. Um ein funktionierendes Spiel zu entwickeln, ist neben der Arbeit in der jeweiligen Gruppe auch zwingend gruppenübergreifende Zusammenarbeit gefordert, z. B. zur Absprache der Schnittstellen und Integration der einzelnen Komponenten. Die Teilnehmerzahl ist aufgrund der Raumgröße und Hardwareverfügbarkeit auf max. 20 Studenten beschränkt. ... (weiter siehe Digicampus)

Prüfung

Vortrag mit Softwarepräsentation, Ausarbeitung mit Softwaredokumentation

Projektarbeit

Modul INF-0198: Intelligente Systeme		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
Lernziele/Kompetenzen: 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.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
Arbeitsaufwand: Gesamt: 150 Std. 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) 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)		
Voraussetzungen: Sinnvoll sind Kenntnisse und Interesse an: Diskreten Strukturen und Logiken		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Intelligente Systeme (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: <ul style="list-style-type: none"> • Wissensrepräsentation, Sprachen, Ontologien (Prädikatenlogik, Hornlogik, Semantische Netze, ...) • Der Inferenzbegriff ("Beweis"kalküle, Resolution) • Klassifikation und Konzepte (Clustering, Hierarchical Clustering, Incremental Clustering) • Konzeptanalyse und Konzepthierarchien (Formal Concept Analysis) • Relationale und Soft-Computing Ansätze zur Datenanalyse (Grobmengen, Fuzzy-Logik, ...) • Generalisierung und Induktion (Subsumption, Negation, Abduktion, ...) • Lernen: Finden von Hypothesen (Induktives Logisches Programmieren, Bagging) 		
Literatur: <ul style="list-style-type: none"> • Eigenes Skriptum • S. Russell & P. Norvig: Artificial Intelligence - A Modern Approach, 3rd Edition, 2010 • weitere Literatur wird in der Vorlesung angegeben 		
Modulteil: Intelligente Systeme (Übung)		
Lehrformen: Übung		
Sprache: Deutsch		
SWS: 2		
Prüfung		
Intelligente Systeme (Klausur) Klausur / Prüfungsdauer: 90 Minuten		

Modul INF-0207: Reinforcement Learning		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul verstehen die Studierenden die wesentlichen Konzepte des Reinforcement Learnings. Sie sind in der Lage, fachliche Lösungskonzepte in Programme und Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Sie verfügen über die Kenntnisse der Denkweise und die Sprache anwendungsrelevanter Disziplinen. Im Rahmen der Vorlesung erlernen sie, lernende Komponenten mit geeigneten Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmen selbstständig zu entwickeln und technisch umzusetzen. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen sowie zum logischen, analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikationen: Fortgeschrittene mathematisch-formale Methodik, Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen, Fertigkeit zur Entwicklung und Umsetzung von Lösungsstrategien für komplexe Probleme, Verstehen von Teamprozessen, Fertigkeit der Zusammenarbeit in Teams, Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen, Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 60 Std. Übung (Präsenzstudium) 120 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)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Reinforcement Learning (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: Markov-Entscheidungsproblem, Dynamische Programmierung, Monte Carlo Methoden, TD-Lernen, Eligibility Traces, Hierarchisches Bestärkendes Lernen, Planen und Lernen, Generalisierung		
Literatur: <ul style="list-style-type: none"> • Skript • Richard S. Sutton und Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998 		
Zugeordnete Lehrveranstaltungen:		
Reinforcement Learning (Vorlesung) Inhalt der Veranstaltung sind Methoden und Prinzipien des Reinforcement Learning. In der Übung werden verschiedene Probleme modelliert, mit Hilfe von Lernalgorithmen als Simulationen implementiert und verglichen. Siehe https://www.informatik.uni-augsburg.de/lehrstuehle/hcm/lectures/2017ws/rl/		

Modulteil: Reinforcement Learning (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Reinforcement Learning (Übung)

Prüfung

Reinforcement Learning (Projektarbeit)

Projektarbeit, Projektarbeit / mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0250: Praktikum Reinforcement Learning		8 ECTS/LP
Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul verstehen die Studierenden die wesentlichen Konzepte des Reinforcement Learnings. 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 zu präsentieren. Sie sind in der Lage, fachliche Lösungskonzepte in Programme und Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Sie verfügen über die Kenntnisse der Denkweise und die Sprache anwendungsrelevanter Disziplinen. Im Rahmen des Praktikums erlernen sie, lernende Komponenten mit geeigneten Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmen selbstständig zu entwickeln und technisch umzusetzen. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen sowie zum logischen, analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikationen: 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>Arbeitsaufwand: Gesamt: 240 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 90 Std. Praktikum (Präsenzstudium)</p>		
<p>Voraussetzungen: Programmiererfahrung Modul Reinforcement Learning (INF-0207) - empfohlen</p>		
<p>Angebotshäufigkeit: jedes Sommersemester</p>	<p>Empfohlenes Fachsemester: ab dem 2.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Praktikum Reinforcement Learning Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		
<p>Inhalte: Die konkrete Aufgabenstellung aus dem Gebiet "Reinforcement Learning" wird jedes Semester neu entworfen.</p>		
<p>Prüfung Vortrag mit Softwarepräsentation, Ausarbeitung mit Softwaredokumentation Projektarbeit</p>		

Modul INF-0272: Intelligente Signalanalyse in der Medizin		5 ECTS/LP
Version 1.1.0 (seit WS17/18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Knowledge: The students learn the principal concepts of sequential signal processing, signal source separation, and feature extraction and information reduction exemplified by medically relevant audio and bio signals. They further gain insight into machine learning principles such as learning dynamics and context as is needed for many intelligent signal analysis tasks. They will learn about different problems and solutions in the analysis of a variety of signals relevant in the context of health care, wellbeing, and general medical signals analysis. Students will get to know the mindset of modern machine learning, computer-aided health care, and get to know ethical implications.</p> <p>Skills: The students will be able to choose appropriate algorithms of signal processing and machine intelligence, further develop these, design new solutions, and apply these to a broad range of medical signal analysis problems. They will practice to think logically and conceptionally in order to select appropriate solutions to a given task. Students will be able to recognise important technical developments in the field of signal processing, machine learning and e-Health/m-Health.</p> <p>Competences: The students are able to characterise, judge on the quality and suitability, and design suited algorithmic solutions for intelligent signal analysis with a focus on medical signals. They are further able to realise the learnt concepts in programs and machine learning models. Participants will be able to analyse and structure complex and practice-oriented problems in the field of m-Health and e-Health and to find suitable and state-of-the-art solutions. They know how to make scientifically meaningful evaluations of proposed systems. They will further learn how to document and present results in a reasonable and meaningful way.</p> <p>Key skills: Formal methods; Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality awareness; Scientific working; Literature research.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std.</p> <p>30 Std. Übung (Präsenzstudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
<p>Voraussetzungen:</p> <p>Knowledge of basic mathematic lectures should be present.</p>		
<p>Angebotshäufigkeit: jedes Sommersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 4</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Intelligente Signalanalyse in der Medizin (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Englisch</p> <p>Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)</p> <p>SWS: 2</p>		

Inhalte:

Topics: Basics of Signal Processing, Signal Source Separation, Data Acquisition and Annotation, Audio-Visual Feature Extraction, Machine Learning, e-Health, m-Health, Ethics, Python, Machine Learning Toolkits.

Literatur:

Björn Schuller: „*Intelligent Audio Analysis*“, Signals and Communication Technology, Springer, ISBN: 978-3642368059, 2013.

Modulteil: Intelligente Signalanalyse in der Medizin (Übung)

Lehrformen: Übung

Sprache: Englisch

Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)

SWS: 2

Prüfung

Intelligente Signalanalyse in der Medizin (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0273: Praktikum Mobile Sensing for Fitness and Well-being		5 ECTS/LP
Version 1.0.0 (seit WS17/18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Students learn to realise concepts and models of sensing health- and fitness-related parameters on smart devices. They learn how to acquire signals from different modalities and sensors and to implement algorithms of pattern recognition on mobile devices. After participation in the Praktikum, students know how to analyse and structure complex problems in the field and to select suitable and state-of-the-art approaches to their solution.</p> <p>Participants are trained in their analytical and conceptional skills as well as in practical programming skills to transfer their knowledge to a practical task. They learn how to make scientifically meaningful assessments of their system using appropriate methods. They will know the mindset of two different fields, software development and machine learning. All knowledge obtained during the Praktikum is applied in practice-oriented tasks. Furthermore, after participation, students will be able to recognise important technical evolution in the field of sensors, mobile apps and related tools, and intelligent signal analysis.</p> <p>Students will work in teams and organise their work and task distribution in an autonomous way. They will learn how to summarise, present and document results in a reasonable way</p> <p>Key skills: Formal methods; Methods for software development and abstraction; Versioning tools; Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams, Understanding of team management; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality Awareness.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Praktikum Mobile Sensing for Fitness and Wellbeing		
Lehrformen: Praktikum		
Sprache: Englisch		
SWS: 4		
<p>Inhalte:</p> <p>Research in the field of m-Health is focussed on the design and development of sensors, systems, and applications to recognise, interpret and simulate human states w.r.t. fitness, health, and wellbeing. In this Praktikum, students will experience in designing relevant systems, which are using modalities originating from different sensors, such as, vital signs, audio, speech, and video. In small teams, they will implement and evaluate an application running on a smart device.</p>		
<p>Literatur:</p> <p>Wird vom Dozenten / von der Dozentin bekannt gegeben</p>		

Zugeordnete Lehrveranstaltungen:

Praktikum Mobile Sensing for Fitness and Wellbeing (Praktikum)

This course will be given as a block course. The schedule and the location are going to be announced later.
Please NOTE that you need to register via STUDIS in order to participate in the Praktikum.

Prüfung

Praktikum Mobile Sensing for Fitness and Wellbeing

Praktikum

Modul INF-0279: Music Informatics		5 ECTS/LP
Version 1.2.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Knowledge: The course Music Informatics presents the fundamental concepts of music theory and the music language and its representation in the visual, symbolic, and acoustic domain. Several digital formats for music symbolic representation, such as Music XML, MEI, Kern**, and MIDI protocol, as well as open source tools such as LilyPond and Csound will be introduced. Machine learning principles and techniques with applications in music information retrieval and computational musicology will be practically applied. Students will learn about different problems and solutions in the analysis of symbolic and acoustic music data. Students will get to know the mindset from both sides, the musicological and the computer scientist perspective.</p> <p>Skills: The students will understand the basic principles of music theory and its representation in digital language, being able to analyse, interpret, and create musical samples in a variety of symbolic formats and programming languages. They will learn to apply machine learning procedures, such as feature extraction and pattern recognition, to music information retrieval problems, such as key detection and music-score synchronisation, amongst other. After participation, students will know how to advance existing concepts and approaches in the field of music informatics and data analysis. Furthermore, they will be able to recognise important technical developments in the field of data science and signal processing.</p> <p>Competences: By integrating basic principles of music theory, its representation in digital language, and machine learning techniques, the students will be able to identify new problems and solutions in the field of music information retrieval considering a variety of musical styles and genres. The students are able to characterise, judge on the quality and suitability, and design suited algorithmic solutions for music data analysis in both the symbolic and the audio domain.</p> <p>Participants will be able to analyse and structure complex and practice-oriented problems in the field of music informatics and to find suitable solutions, by using state-of-the-art tools and complementary methods, if needed. They know how to make scientifically meaningful evaluations of proposed systems. They will further learn how to document and present results in a reasonable and meaningful way.</p> <p>Key skills: Computational musicology, Music theory, Digital Music Representation, Basics of Signal Processing, Machine Learning, Music Information Retrieval, Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality awareness; Scientific working; Literature research.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std.</p> <p>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> <p>60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p>		
Voraussetzungen:		
Knowledge of basic mathematic lectures should be present		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	

Moduleile
Moduleil: Music Informatics (Lecture) Lehrformen: Vorlesung Sprache: Englisch SWS: 2
Inhalte: In Music Informatics, the basic principles of music theory will be presented from both the traditional and computational point of view. Music will be evaluated in three domains: visual, symbolic, and acoustic; and for each of them: formats, programming languages, and machine learning tools will be studied. This course will give a basic introduction to music information retrieval and computational musicology by identify problems and solutions for different kinds of musical genres and styles.
Literatur: <ul style="list-style-type: none">• Meinard Müller: "<i>Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications.</i>" Springer, ISBN: 978-3-319-21944-8. 2015.• Björn Schuller: "<i>Intelligent Audio Analysis</i>", Signals and Communication Technology, Springer, ISBN: 978-3642368059, 2013.
Moduleil: Music Informatics (Tutorial) Lehrformen: Übung Sprache: Englisch SWS: 2
Prüfung Music Informatics (Exam) Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0284: Praktikum Mobile Application Development		5 ECTS/LP
Version 1.0.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Students learn to realise concepts and models of mobile application development on the Android platform. They learn how to acquire data and signals from different sensors and to implement algorithms of pattern recognition and data/signal analysis on mobile devices. After participation in the Praktikum, students know how to analyse and structure complex problems in the field and to select suitable and state-of-the-art approaches to their solution.</p> <p>Participants are trained in their analytical and conceptional skills as well as in practical programming skills to transfer their knowledge to a practical task. They learn how to make scientifically meaningful assessments of their system using appropriate methods. They will know the mindset and thinking of application development and software engineering. All knowledge obtained during the Praktikum is applied in practice-oriented tasks. Furthermore, after participation, students will be able to recognise important technical evolution in the field of sensors, mobile apps and related tools, and smart devices.</p> <p>Students will work in teams and organise their work and task distribution in an autonomous way. They will learn how to summarise, present and document results in a reasonable way.</p> <p>Key skills: Formal methods; Methods for software development and abstraction; Versioning tools; Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams, Understanding of team management; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality Awareness.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
Voraussetzungen:		
Programming skills in Java are required.		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Praktikum Mobile Application Development</p> <p>Lehrformen: Praktikum</p> <p>Sprache: Englisch</p> <p>SWS: 4</p>
<p>Inhalte:</p> <p>In the Praktikum Mobile Application Development, students will develop software for the Android platform in groups of 2-4 students.</p> <p>Students will get to know and to program several groups of Android devices, e.g., mobiles and wearables, and respective sensors. They will familiarise with the Android Studio IDE, the according toolchain and the code management tool Gi.</p> <p>Topics: Java, Android, App development, UI design, Audio Player, Map and Location, Camera control, Mobile devices, Intelligent signal processing, Wearables.</p>

Literatur:

The literature is going to be announced by the lecturers during the course.

Prüfung

Praktikum Mobile Application Development

Praktikum

Modul INF-0293: Advanced Deep Learning <i>Advanced Deep Learning</i>		8 ECTS/LP
Version 1.0.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: Nach der Teilnahme an dem Praxismodul verfügen die Studierenden über detailliertes und aktuelles Fachwissen auf dem Gebiet des maschinellen Lernens, können bedeutsame technische Entwicklungen identifizieren und können eine komplette Pipeline zur multimodalen Datenverarbeitung mit tiefen neuronalen Netzen implementieren. Sie können Problemstellungen und Ergebnisse des Gebiets präzise beschreiben, diskutieren und gelernte Konzepte und Methoden auf ähnliche Problemstellungen im maschinellen Lernen anwenden. Darüber hinaus analysieren die Studierenden weiterführende Konzepte, Methoden, Verfahren, Techniken und Technologien aus dem Gebiet des maschinellen Lernens, um sie in Forschungsprojekten anzuwenden, diese auf aktuelle industriennahe Aufgabenstellungen zu übertragen und dort aktiv mitzuarbeiten. Die Studierenden lernen wissenschaftlich anspruchsvolle Themenstellungen im Gebiet des maschinellen Lernens auf andere Forschungsfragen zu übertragen und darauf aufbauend ein komplexes Projekt in Gruppenarbeit auszuarbeiten. Außerdem verfügen sie über die Team- und Kommunikationsfähigkeit, um Problemstellungen auf dem Gebiet zu erörtern, Fragen und Zwischenergebnisse zu diskutieren, beschreiben und zu präsentieren. Zusätzlich sind die Studierenden in der Lage detaillierte Experimente durchzuführen und Ergebnisse zu beurteilen, vergleichen und auf Plausibilität zu prüfen.</p> <p>Schlüsselqualifikationen: Fortgeschrittene mathematisch-formale Methodik; Umsetzen fachlicher Lösungskonzepte in Programme und Modelle; Methoden zur Entwicklung größerer Softwaresysteme, Konstruktion von Abstraktionen und Architekturen; Fachübergreifende Kenntnisse; Systematische Weiterentwicklung von Entwurfsmethoden; Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten; Verstehen von Teamprozessen; Fertigkeit der Zusammenarbeit in Teams; Fähigkeit zur Leitung von Teams; Kennenlernen der Abläufe und Prozesse im Anwendungsumfeld der Informatik; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen; Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 20 Std. Vorlesung (Präsenzstudium) 40 Std. Übung (Präsenzstudium)</p>		
<p>Voraussetzungen: Grundkenntnisse in maschinellem Lernen und maschinellem Sehen (Grundstudiums Vorlesungen Multimedia Grundlagen 1 + 2 und Master-Vorlesung "Multimedia 2 - Machine Learning and Computer Vision") Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia II: Machine Learning and Computer Vision (INF-0092) - empfohlen</p>		<p>ECTS/LP-Bedingungen: Bestehen der Portfolioprüfung</p>
<p>Angebotshäufigkeit: jedes Semester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	

Modulteile
Modulteil: Advanced Deep Learning (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2
Inhalte: <ul style="list-style-type: none">• Deep Learning in general• Deep Convolutional Neural Networks• Transfer Learning• Recurrent Neural Networks / LSTM Networks• Natural Language Processing• Multimodal Fusion (Vision+Language)• Application: Image Captioning
Zugeordnete Lehrveranstaltungen: Advanced Deep Learning (Vorlesung)
Modulteil: Advanced Deep Learning (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 4
Zugeordnete Lehrveranstaltungen: Übung zu Advanced Deep Learning (Übung)
Prüfung Advanced Deep Learning Portfolioprüfung, Die Endnote setzt sich aus bewerteten Übungsblättern und einem bewerteten Teamprojekt zusammen.

Modul INF-0294: Speech Pathology		5 ECTS/LP
Version 1.0.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Knowledge: The students learn concepts relating to signal processing, speech production, phonetics, speech and language pathology, speech analysis, feature extraction, denoising and information reduction as exemplified through the analysis of automated voice pathology detection. They further gain insight into machine learning principles, with a particular focus on deep learning solutions, as is needed to diagnose a range of different voice pathologies. They will learn about different problems and solutions in the analysis of a variety of speech, relevant in the context of health care and wellbeing.</p> <p>Skills: The students will be able to choose appropriate algorithms of signal processing and machine intelligence, further develop these, design new solutions, and apply these to the task of voice pathology detection. They will know how to analyse and structure complex problems in the field, to employ suitable approaches to solve them, and to transfer knowledge to similar tasks. After participation in the course, they will be able to implement approaches and models into programs. Students will be able to assess developed systems in a scientific way. Important technical evolution and novelties in the fields of speech analysis and medical machine learning will be recognised by them.</p> <p>Competences: The students are able to characterise, judge on the quality and suitability, and design suited algorithmic solutions for intelligent signal analysis with a focus on voice pathology detection. They are further able to present and document results in a reasonable and meaningful way.</p> <p>Key skills: Formal methods; Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality awareness; Scientific working; Literature research.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std.</p> <p>60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p>		
<p>Voraussetzungen:</p> <p>Knowledge of basic mathematic lectures should be present.</p>		
<p>Angebotshäufigkeit: jedes Wintersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 4</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Speech Pathology (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Englisch</p> <p>SWS: 2</p>		

Inhalte:

The course "Pathological Speech" will give an introduction to models of speech production (e.g., source-filter models) with a focus on aspects that are relevant to pathologies and their recognition using automated methods of signal processing and machine learning. Moreover, students learn about robust feature extraction, modern methods of machine learning and machine intelligence, and implementation of such systems on devices

Topics: Speech production; Phonetics; Speech and language pathology; Signal processing; Natural language processing; Speech analysis; Feature extraction; Machine learning; Deep learning; Denoising; Information reduction; Healthcare.

Literatur:

- Björn Schuller, Anton Batliner: "Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing", Wiley, ISBN: 978-1119971368, 2013.
- Further literature is going to be announced during the lecture.

Zugeordnete Lehrveranstaltungen:

Speech Pathology (Vorlesung)

In this novel course, students will learn about human speech production and related pathologies. The course will introduce recent source-filter models and the principal concepts of signal processing, feature extraction, and information reduction. State-of-the-art machine learning techniques will be discussed as well as implications of practical applications of speech-related health care technologies.

Modulteil: Speech Pathology (Übung)

Lehrformen: Übung

Sprache: Englisch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Speech Pathology (Übung)

Prüfung

Speech Pathology

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0296: Praktikum Interactive Machine Learning		8 ECTS/LP
Version 1.0.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul sind die Studierenden mit Grundkonzepten des interaktiven maschinellen Lernens vertraut. Sie sind in der Lage, fachliche Lösungskonzepte in Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Darüber hinaus erhalten sie einen Einblick in aktuelle Arbeiten des Forschungsgebiets. Weiterhin werden Kompetenzen in den Bereichen Team- und Kommunikationsfähigkeit sowie Selbstorganisation durch die Bearbeitung von Projektaufgaben und deren eigenständige Projektplanung vermittelt. Im Rahmen dieser Projektaufgaben werden Kenntnisse der Denkweise und Sprache des Forschungsgebietes durch die Diskussion und Präsentation der Projektergebnisse gefördert.</p> <p>Schlüsselqualifikationen: Umsetzen fachlicher Lösungskonzepte in Programme und Modelle; Fachspezifische Vertiefung; Fachübergreifende Kenntnisse; Fertigkeit, mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen anzustellen; Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen; Fertigkeit der Zusammenarbeit in Teams; Kenntnisse von praxisrelevanten Aufgabenstellungen; Kennenlernen der Abläufe und Prozesse im Anwendungsumfeld der Informatik; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p>Voraussetzungen: Programmiererfahrung</p>		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Praktikum Interactive Machine Learning Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>
<p>Zugeordnete Lehrveranstaltungen: Interactive Machine Learning (Praktikum) Dieser Kurs fokussiert sich auf "interaktives Maschinelles Lernen" (iML). Wir definieren iML als Maschinelles Lernen bei dem der Mensch als zentraler Teil, interaktiv an verschiedenen Stellen in den Lernprozesses eingreift. Er überwacht dabei die Ergebnisse der Maschine und liefert Eingaben und Korrekturen um den Lernprozess zu verbessern. Während traditionelle ML Systeme menschliche Eingaben auf das Bereitstellen von Annotationen beschränken, beschäftigt sich iML damit Interaktionsdesign Richtlinien für ML Systeme zu erschaffen, und neue Methoden zu entwickeln menschliche Expertise in den ML Prozess einfließen zu lassen. Dadurch werden Systeme transparenter und nachvollziehbarer, was sich mit dem aktuell aufstrebenden Forschungstrend der "explainable AI" (XAI) überschneidet. Der Kurs besteht aus mehreren praktischen Hands-on Projekten und einer Einführung in verschiedene Themen rund um das Thema "interactive Machine Learning". Grundkenntnisse in C/C++ oder Python sind erwünscht. ... (weiter siehe Digicampus)</p>

Prüfung

Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation

Projektarbeit

Modul INF-0033: Modellgetriebene Softwareentwicklung		6 ECTS/LP
Version 2.0.0 (seit WS14/15) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Modellgetriebene Softwareentwicklung oder Model Driven Software Development (MDSD) befasst sich mit der Effizienzsteigerung in der Softwareherstellung durch Automatisierung und Wiederverwendung. Die Teilnehmer lernen in der Veranstaltung Methoden zur modellgetriebenen Entwicklung von Softwaresysteme anzuwenden und zu vergleichen. Sie entwickeln dabei vertiefte, fachspezifische Lösungskonzepte für MDSD. Sie können aktuelle Technologien und Standards für MDSD bewerten und analysieren deren Anwendbarkeit in praxisrelevanten Aufgabenstellungen. Die Teilnehmer bauen Fertigkeiten zur Analyse und Strukturierung komplexer Informatikproblemstellungen bei der Generierung von Infrastrukturcode, Subsysteme, Konfigurationen oder ganze Anwendungen aus Modellen auf. Dabei entwickeln sie Fertigkeiten zum logischen, analytischen und konzeptionellen Denken und können Problemlösungen systematisch entwickeln und beurteilen.</p> <p>Schlüsselqualifikation: Fachübergreifende Kenntnisse; Kompetenz zur Vernetzung unterschiedlicher Fachgebiete; Team- und Kommunikationsfähigkeit; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein; Kennenlernen der Abläufe und Prozesse im Anwendungsumfeld der Informatik; Formale quantitative Grundlagen kennen und verstehen; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 30 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 5	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Modellgetriebene Softwareentwicklung (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 3		
Inhalte: Modellgetriebene Softwareentwicklung oder Model Driven Software Development (MDSD) befasst sich mit der Effizienzsteigerung in der Softwareherstellung durch Automatisierung und Wiederverwendung. Dabei werden Infrastrukturcode, Subsysteme, Konfigurationen oder ganze Anwendungen aus Modellen generiert.		
<p>Literatur:</p> <ul style="list-style-type: none"> • Folien • Pohl et al. Software Product Line Engineering: Foundations, Principles, and Techniques • Kleppe et al: MDA explained • Hitz et al: UML@Work • weitere Literatur in der Vorlesung zu speziellen Themen 		

Modulteil: Modellgetriebene Softwareentwicklung (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Modellgetriebene Softwareentwicklung

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0034: Softwarearchitekturen und Enterprise Architecture Management		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Die Studierenden sind in der Lage SW-Architekturen zu erstellen (K3), zu bewerten (K6) und zu dokumentieren. Dafür können sie fachliche Lösungskonzepte in Modelle überführen und kennen Methoden zur Entwicklung solcher Abstraktionen und Architekturen. Sie können Vor- und Nachteile von Entwurfsalternativen beschreiben (K4) und können diese im jeweiligen Anwendungszusammenhang bewerten (K6). Probleme können dabei eigenständig erkannt (K4) und Lösungen systematisch entworfen (K5) und realisiert (K3) werden. Weiterhin haben sie Fertigkeiten für die Analyse und Strukturierung der Problemstellungen in Unternehmensarchitekturen entwickelt und kennen die Konzepte und Vorgehensweisen für die Erstellung solcher Architekturen. Die Studierenden können praxisrelevanten Fragestellungen in Unternehmensarchitekturen benennen (K1). Sie können geeignete Methoden zur Architekturerstellung und -bewertung auswählen und sicher anwenden. Die Studenten kennen Modellierungssprachen und Patterns zur Erstellung von Software- und Unternehmensarchitekturen. Sie haben die Kompetenz bedeutende technische Entwicklungen zu erkennen.</p> <p>Schlüsselqualifikation: Kompetenz zur Vernetzung unterschiedlicher Fachgebiete; Team- und Kommunikationsfähigkeit; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Praxiserfahrung und Berufsbefähigung</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 45 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
<p>Voraussetzungen: Frühere Veranstaltung "Softwarearchitekturen und Technologien für eingebettete Systeme" darf nicht belegt worden sein wegen Überschneidungen.</p>		
<p>Angebotshäufigkeit: jedes Wintersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 5</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Softwarearchitekturen und Enterprise Architecture Management (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 3</p> <p>Inhalte: Der Vorlesungsinhalt umfasst Patterns, Modellierungstechniken und die Evaluation von Softwarearchitekturen. Weiterhin wird auf den Bereich des Enterprise Architecture Managements eingegangen.</p> <p>Literatur:</p> <ul style="list-style-type: none"> • Bass et al: Software Architecture in Practice • Clements et al: Documenting Software Architectures • Clements et al: Evaluation of Software Architectures • Kopetz: Real-Time Systems 		

Zugeordnete Lehrveranstaltungen:

Softwarearchitekturen und EAM (Vorlesung)

Inhalte: - Was sind SW-Architekturen? - Pattern und Muster für SW-Architekturen - Modellierung von SW-Architekturen - Evaluation von SW-Architekturen - Eingebettete Systeme: Definitionen, Anforderungsanalyse, Modellierung, Architektur - Software-Qualität: Definitionen und Standards, Funktionstest, Überdeckungsmaße, HiL-, Integrations- und Abnahmetests, Verifikation und Validierung, Architecture Design and Reliability - Enterprise Architecture Management: Methoden, Frameworks, Tools

Modulteil: Softwarearchitekturen und Enterprise Architecture Management (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Softwarearchitekturen und EAM (Übung)

Der Übungsbetrieb soll die Inhalte der Vorlesung veranschaulichen und wird aus der gruppenweisen Vorstellung praktischer Beispiele von Architekturen und Frameworks bestehen. Die konkreten Themen folgen.

Prüfung

Softwarearchitekturen und Enterprise Architecture Management (mündl. Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0037: Praktikum Automotive Software Engineering		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Praktikum Automotive Software Engineering verstehen die Studierenden praxisnaher Problemstellungen höherer Komplexität im Bereich der Entwicklung und Absicherung von Fahrassistenzsystemen mit aktuellen Methoden und Tools der modellbasierten Entwicklung eingebetteter Systeme. Die Studierenden erlangen tiefergehende fachspezifische als auch fächerübergreifende Kenntnisse und Fähigkeiten, beispielsweise aus der Regelungstechnik, Fahrphysik und Mathematik. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren, zu präsentieren und verständlich zu dokumentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; Verständliche, sichere und überzeugende Präsentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein; Kommunikationsfähigkeit; Fertigkeit der Zusammenarbeit in Teams und Verstehen von Teamprozessen; Projektmanagementfähigkeiten</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 90 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p>Voraussetzungen: 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</p>		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Praktikum Automotive Software Engineering (MA) Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		

Inhalte:

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.

Literatur:

abhängig vom Thema

Zugeordnete Lehrveranstaltungen:

Praktikum zu Automotive Software Engineering (Praktikum)

Im Automotive-Praktikum lernen die Teilnehmer/innen wie Software für Automotive-Anwendungen modellbasiert entwickelt, simuliert und analysiert werden kann. In einem Einführungskurs werden die notwendigen Grundlagen anhand von eigens dafür konzipierten Tutorials erlernt. Die Teilnehmer/innen lernen dabei u.a. die im Automotive-Umfeld häufig eingesetzte Modellierungswerkzeug-Kombination „MATHLAB/Simulink“ sowie das graphische Simulations- und Analyse-Tool „CarMaker“ kennen. Im Anschluss an den Einführungskurs werden in Gruppen von drei bis vier Teilnehmer/innen Übungsaufgaben zur Vertiefung der im Einführungskurs erlernten Kenntnisse bearbeitet. Dies garantiert eine optimale Vorbereitung an die anschließende Projektbearbeitung. Im Rahmen des Projektes werden in Teamarbeit Fahrerassistenzsysteme oder Teilaspekte des autonomen Fahrens modelliert, implementiert und getestet. In einer abschließenden Präsentation werden die Ergebnisse vorgestellt und diskutiert. Die Arbeitsunterlagen (Tutoria ... (weiter siehe Digicampus)

Prüfung

Praktikum Automotive Software Engineering (MA)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0038: Praktikum Avionic Software Engineering	6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer	
<p>Inhalte:</p> <p>Die Teilnehmer im Avionik-Praktikum erlernen, wie Software für komplexe Avionik-Systeme entwickelt wird. In kleinen Gruppen wird von den Studenten ein einfacher Autopilot für ein fliegendes System umgesetzt und in einer Simulationsumgebung getestet. Die so entwickelte Steuerung soll anschließend auf einem realen Quadrocopter zum Einsatz kommen. Die Studierenden erarbeiten anhand der Aufgabenstellung eine Software-Architektur, und realisieren diese innerhalb eines Frameworks zur Anbindung an die Simulationsumgebung (X-Plane).</p> <p>In einer mehrtägigen Einführungs-Blockveranstaltung erwerben die Teilnehmer die nötigen Grundkenntnisse über die Entwicklung zuverlässiger Avionik-Systeme und erhalten einen Überblick über die für dieses Praktikum verwendeten Konzepte und Technologien:</p> <ul style="list-style-type: none"> • Techniken zur Entwicklung sicherheitskritischer Systeme • Qualitätssicherung in der Software-Entwicklung durch den Einsatz geeigneter Werkzeuge • Einführung in die Steuerung fliegender Systeme und Navigation • Einführung in die Physik fliegender Systeme • Komponentenbasierte Software-Entwicklung mit Java und OSGi • Echtzeitfähige Software in Java gemäß der RTSJ-Spezifikation <p>Das Praktikum wird in den Semesterferien angeboten und besteht aus dem theoretischen Teil als Blockveranstaltung und der anschließenden mehrwöchigen selbstständigen Umsetzung der Praktikumsaufgabe durch die Studierenden.</p> <p>Die erforderlichen Tätigkeiten sind:</p> <ul style="list-style-type: none"> • Erstellung einer geeigneten Software-Architektur und -Design • Implementierung eines grundlegenden Autopiloten innerhalb des vorgebenen Frameworks in Java und OSGi • Entwicklung von Software-Komponenten zur Steuerung der Flughardware • Überprüfung der funktionalen Korrektheit durch Unit- und Integrationstests 	
<p>Lernziele/Kompetenzen:</p> <p>Nach der Teilnahme am Praktikum Avionic Software Engineering verfügen die Studierenden über detailliertes und aktuelles Wissen auf dem Gebiet der Softwareentwicklung komplexer Systeme im Bereich der Luftfahrt. Sie können Problemstellungen und Ergebnisse des Gebiets präzise beschreiben und vergleichen. Darüber hinaus verstehen die Studierenden tiefergehende Konzepte, Methoden, Verfahren, Techniken und Technologien aus dem genannten Gebiet, um eigenene Lösungsansätze zu planen und die erlernten Kompetenzen bei der Umsetzung aktiv anzuwenden. Durch die Lösung der praktischen Aufgabenstellung in Gruppen entwickeln sie die Team- und Kommunikationsfähigkeit, um Problemstellungen auf dem Gebiet zu erörtern, Fragen und Zwischenergebnisse zu diskutieren und die gemeinsam ausgearbeiteten Lösungen zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Recherche in (auch englischsprachlichen) Lehrbüchern, Handbüchern und Dokumentationen; Verständliche, sichere und überzeugende Präsentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein; Kommunikationsfähigkeit; Projektmanagementfähigkeiten; Fertigkeit der Zusammenarbeit in Teams und Verstehen von Teamprozessen; Kenntnisse von praxisrelevanten Aufgabenstellungen</p>	
<p>Arbeitsaufwand:</p> <p>Gesamt: 180 Std.</p> <p>90 Std. Praktikum (Präsenzstudium)</p> <p>90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>	
<p>Voraussetzungen:</p> <p>Grundkenntnisse im Bereich Software Engineering; Programmiererfahrung in Java; Interesse an Avionik-Systemen; Keine Erfahrung mit OSGi erforderlich</p>	

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		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Moduleile
<p>Modulteil: Praktikum Avionic Software Engineering</p> <p>Lehrformen: Praktikum</p> <p>Sprache: Deutsch</p> <p>SWS: 6</p>
<p>Inhalte:</p> <p>Die Teilnehmer im Avionik-Praktikum erlernen, wie Software für komplexe Avionik-Systeme entwickelt wird. In kleinen Gruppen wird von den Studenten ein einfacher Autopilot für ein fliegendes System umgesetzt und in einer Simulationsumgebung getestet. Die so entwickelte Steuerung soll anschließend auf einem realen Quadrocopter zum Einsatz kommen. Die Studierenden erarbeiten anhand der Aufgabenstellung eine Software-Architektur, und realisieren diese innerhalb eines Frameworks zur Anbindung an die Simulationsumgebung (X-Plane).</p> <p>In einer mehrtägigen Einführungs-Blockveranstaltung erwerben die Teilnehmer die nötigen Grundkenntnisse über die Entwicklung zuverlässiger Avionik-Systeme und erhalten einen Überblick über die für dieses Praktikum verwendeten Konzepte und Technologien:</p> <ul style="list-style-type: none"> • Techniken zur Entwicklung sicherheitskritischer Systeme • Qualitätssicherung in der Software-Entwicklung durch den Einsatz geeigneter Werkzeuge • Einführung in die Steuerung fliegender Systeme und Navigation • Einführung in die Physik fliegender Systeme • Komponentenbasierte Software-Entwicklung mit Java und OSGi • Echtzeitfähige Software in Java gemäß der RTSJ-Spezifikation <p>Das Praktikum wird in den Semesterferien angeboten und besteht aus dem theoretischen Teil als Blockveranstaltung und der anschließenden mehrwöchigen selbstständigen Umsetzung der Praktikumsaufgabe durch die Studierenden.</p> <p>Die erforderlichen Tätigkeiten sind:</p> <ul style="list-style-type: none"> • Erstellung einer geeigneten Software-Architektur und -Design • Implementierung eines grundlegenden Autopiloten innerhalb des vorgebenen Frameworks in Java und OSGi • Entwicklung von Software-Komponenten zur Steuerung der Flughardware • Überprüfung der funktionalen Korrektheit durch Unit- und Integrationstests
<p>Literatur:</p> <p>abhängig vom Thema</p>
<p>Zugeordnete Lehrveranstaltungen:</p> <p>Praktikum Avionic Software Engineering (Praktikum)</p> <p>Die Teilnehmer im Avionik-Praktikum erlernen, wie Software für komplexe Avionik-Systeme entwickelt wird. In kleinen Gruppen wird von den Studenten ein einfacher Autopilot für ein fliegendes System umgesetzt und in einer Simulationsumgebung getestet. Die so entwickelte Steuerung soll anschließend auf einem realen Quadrocopter zum Einsatz kommen. Die Studierenden erarbeiten anhand der Aufgabenstellung eine Software-Architektur, und realisieren diese innerhalb eines Frameworks zur Anbindung an die Simulationsumgebung (X-Plane). In einer mehrtägigen Einführungs-Blockveranstaltung erwerben die Teilnehmer die nötigen Grundkenntnisse über</p>

die Entwicklung zuverlässiger Avionik-Systeme und erhalten einen Überblick über die für dieses Praktikum verwendeten Konzepte und Technologien: • Techniken zur Entwicklung sicherheitskritischer Systeme • Qualitätssicherung in der Software-Entwicklung durch den Einsatz geeigneter Werkzeuge • Einführung in die Steuerung fliegender Systeme und Navigation • Einf
... (weiter siehe Digicampus)

Prüfung

Praktikum Avionic Software Engineering

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0051: Algorithmen für NP-harte Probleme		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: Kenntnis verschiedener algorithmischer Lösungsansätze für NP-harte Probleme und die Fähigkeit, diese sinnvoll im Kontext neuer Probleme einzusetzen, zu analysieren und zu bewerten.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.		
Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 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)		
Voraussetzungen: Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes, insbesondere im Bereich der Graphenalgorithmen. Modul Informatik 3 (INF-0111) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Algorithmen für NP-harte Probleme (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4		
Inhalte: 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.		
Literatur: <ul style="list-style-type: none"> • Skript • Ausiello et al., Complexity and Approximation, Springer, Berlin, 1999. • J. Hromkovic, Algorithmics for Hard Problems, Springer, Berlin, 2001. 		
Modulteil: Algorithmen für NP-harte Probleme (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 2		

Prüfung

Algorithmen für NP-harte Probleme (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0052: Einführung in die Komplexitätstheorie		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: Verständnis für zentrale Themen und Methoden der Komplexitätstheorie; die Fähigkeit, komplexitätstheoretische Sachverhalte und Fragen präzise zu formulieren und zu untersuchen.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.		
Arbeitsaufwand: 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)		
Voraussetzungen: 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		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Einführung in die Komplexitätstheorie (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2		
Inhalte: 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.		
Literatur: <ul style="list-style-type: none"> • Skript • Christos H. Papadimitriou, Computational Complexity, Addison-Wesley, Reading, Mass., 1994. 		
Modulteil: Einführung in die Komplexitätstheorie (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 2		

Prüfung

Einführung in die Komplexitätstheorie (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0053: I/O-effiziente Algorithmen		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: 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.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.		
Arbeitsaufwand: 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)		
Voraussetzungen: Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes. Modul Informatik 3 (INF-0111) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: I/O-effiziente Algorithmen (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: 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".		
Literatur: <ul style="list-style-type: none"> • Skript • J.S. Vitter, Algorithms and data structures for external memory, Foundations and Trends in Theoretical Computer Science 2 (2008), pp. 305-474 		

Modulteil: I/O-effiziente Algorithmen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

I/O-effiziente Algorithmen (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Beschreibung:

Die Prüfung findet zum Ende des Semesters statt. Es besteht bei Bedarf die Möglichkeit einer Nachholprüfung im Folgesemester.

Modul INF-0054: Datenstrukturen		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: Kenntnis einer Reihe von nichtelementaren Datenstrukturen und ihrer Analyse; Fähigkeit zur Anpassung dieser Datenstrukturen an neue Anwendungen und zur Entwicklung neuer einfacher Datenstrukturen.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.		
Arbeitsaufwand: 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)		
Voraussetzungen: Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes. Modul Informatik 3 (INF-0111) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
Modulteil: Datenstrukturen (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4
Inhalte: 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.
Literatur: Skript
Zugeordnete Lehrveranstaltungen: Datenstrukturen (Vorlesung) 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.

Modulteil: Datenstrukturen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Datenstrukturen (Übung)

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.

Prüfung

Datenstrukturen (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0056: Online-Algorithmen		5 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: 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.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.		
Arbeitsaufwand: 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)		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Online-Algorithmen (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2		
Inhalte: 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.		
Literatur: <ul style="list-style-type: none"> • Skript • A. Borodin und R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998. 		
Modulteil: Online-Algorithmen (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 2		

Prüfung

Online-Algorithmen (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0066: Organic Computing II		5 ECTS/LP
Version 1.2.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen: Teilnehmer sind in der Lage, wesentliche Konzepte und Methoden des Organic Computing (OC) wiederzugeben. Dazu zählen unter anderem: Selbst-Organisation, Selbst-Adaption, Robustheit, Flexibilität, Observer/Controller-Architekturen (O/C), Selbst-X Eigenschaften, Extended Classifier Systems (XCS), Genetische Algorithmen (GA), Partikel Swarm Optimization, Influence Detection und Trust. Sie können zudem begründen, wieso es sinnvoll ist OC-Systeme zu betrachten und praxisrelevante Beispiele nennen, in welchen OC-Techniken angewendet werden sollten. Die Teilnehmer sind befähigt, größere Softwaresysteme mit Hilfe der O/C-Architektur zu entwickeln, sowie diese mit passenden OC-Techniken zu befüllen und sich, falls erforderlich, neue OC-Techniken anzueignen ein gegebenes Problem zu lösen. Die Studenten haben die Fähigkeit zum Lösen von praktischen Aufgaben mit Hilfe des XCS, dem Swarming und einem GA. Sie können zudem ein Konzept für ein OC-System ausarbeiten und beurteilen, sowie wissenschaftliche Bewertungen im Bezug auf OC-Systeme durchführen.</p> <p>Schlüsselqualifikationen: Fähigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen, Fähigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten., Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen, Fertigkeit der Zusammenarbeit in Teams.</p>		
<p>Arbeitsaufwand: 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)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Organic Computing II (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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 naturanaloger 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>		

Literatur:

- Folien
- Müller-Schloer und Tomforde: Organic Computing - Technical Systems for Survival in the Real World, Springer International Publishing, 2018, ISBN 978-3-319-68477-2
- Müller-Schloer et al.: Organic Computing - A Paradigm Shift for Complex Systems, Birkhäuser Verlag, Basel, 2011, ISBN 978-3034801294
- Würtz (ed.): Organic Computing (Understanding Complex Systems), SpringerVerlag Berlin, 2008, ISBN 978-3540776567
- Mitchell: Machine Learning, The McGraw-Hill Companies, 1997, ISBN 978-0071154673
- Goldberg: Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989, ISBN 978-0201157673
- Michalewicz, Fogel: How to Solve it: Modern Heuristics, Springer Verlag Berlin, 2004, ISBN 978-3540224945
- Tomforde: Runtime Adaptation of Technical Systems, Südwestdeutscher Verlag für Hochschulschriften, 2012, ISBN 978-3838131337

Modulteil: Organic Computing II (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

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.

Modul INF-0067: Peer-to-Peer und Cloud Computing		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen: Die erfolgreiche Teilnahme an dieser Lehrveranstaltung resultiert in fachlichen Kompetenzen in zwei Teilbereichen: Peer-to-Peer-Systeme und Cloud-Computing.</p> <p>Peer-to-Peer-Systeme Die Studierenden können verschiedene Peer-to-Peer-System-Architekturen und deren Vor- und Nachteile beschreiben. Insbesondere sind sie in der Lage, für einen gegebenen Anwendungsfall eine passende Architektur auszuwählen und diese Entscheidung zu begründen. Des Weiteren können sie gängige dezentrale Suchalgorithmen und -heuristiken einordnen und bezüglich ihrer Effizienz vergleichen. Sie verstehen abstrakte Netzwerkmodelle, und können mit deren Hilfe Peer-to-Peer-Netzwerke analysieren.</p> <p>Cloud-Computing Teilnehmer können die dem Cloud-Computing zugrunde liegenden Technologien unterscheiden und basierend auf diesem Wissen Cloud-Computing-Architekturen analysieren. Sie kennen die Wichtigkeit von Lastverteilung im Rahmen von großen Cloud-Computing-Anwendungen und können die verschiedenen Ansätze dafür erklären und gegeneinander abwägen. Die Studierenden kennen moderne Ansätze zu nebenläufigen Berechnungen über große Datenmengen und können diese auf einfache Probleme anwenden. Sie sind in der Lage, einfache Anwendungen mithilfe von Platform-as-a-Service-Frameworks zu entwickeln.</p> <p>Schlüsselqualifikationen Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen.</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Peer-to-Peer und Cloud Computing (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		

<p>Inhalte:</p> <p>Dieses Modul vermittelt Ansätze zur Organisation von komplexen abstrahierten IT-Infrastrukturen, die dynamisch an wechselnde Nutzungsbedingungen angepasst werden können und Dienste wie Rechenkapazität, Datenspeicher, Netzkapazitäten oder Software auf verschiedenen Ebenen zur Verfügung stellen. Dazu werden die Anforderungen, Eigenschaften 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.</p>
<p>Literatur:</p> <ul style="list-style-type: none"> • aktuelle wissenschaftliche Paper • Mahlmann und Schindelhauer: Peer-to-Peer Netzwerke - Algorithmen und Methoden, Springer 2007 • Antonopoulos und Gillam: Cloud Computing - Principles, Systems and Applications, Springer 2010
<p>Zugeordnete Lehrveranstaltungen:</p> <p>Peer-to-Peer und Cloud Computing (Vorlesung)</p> <p>Die Vorlesung behandelt u.a. folgende Themen: • Netzwerkmodelle • strukturierte Peer-to-Peer-Systeme – Distributed Hash Tables – Chord – CAN • unstrukturierte Peer-to-Peer-Systeme – Suche – Caching und Replikation – Technologien der verschiedenen Gnutella-Netzwerke • Cloud Computing–Technologien – Virtualisierung – Service-orientierte Architekturen – Cloud-Architekturen – Load Balancing–Ansätze</p>
<p>Modulteil: Peer-to-Peer und Cloud Computing (Übung)</p> <p>Lehrformen: Übung</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p>
<p>Zugeordnete Lehrveranstaltungen:</p> <p>Übung zu Peer-to-Peer und Cloud Computing (Übung)</p> <p>Die zum Modul „Peer-to-Peer und Cloud Computing“ zugehörige Übung bietet die Möglichkeit, die in der Vorlesung erlernten Konzepte zu vertiefen und beispielhaft anzuwenden. Dazu gehören u.a. die Analyse von wissenschaftlichen Beiträgen zu einigen Schwerpunkten der Vorlesung sowie das Lösen anwendungsorientierter Aufgaben.</p>
<p>Prüfung</p> <p>Peer-to-Peer und Cloud Computing (mündliche Prüfung)</p> <p>Mündliche Prüfung / Prüfungsdauer: 30 Minuten</p> <p>Beschreibung:</p> <p>Die Prüfung kann jedes Semester zu Beginn und Ende der vorlesungsfreien Zeit abgelegt werden.</p>

Modul INF-0069: Weiterführende Betriebssystemkonzepte		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen: Die Veranstaltung vermittelt den Studierenden bei erfolgreicher Teilnahme fundierte fachliche Kenntnisse über moderne Betriebssysteme und die dahinterstehenden grundlegenden Konzepte. Sie können danach die Geschichte der Betriebssysteme wiedergeben und deren primäre Typen erläutern. Der theoretische Teil konzentriert sich auf die Vertiefung des Verständnisses von Mechanismen aus den Bereichen Multiprogramming, Interprozesskommunikation, Speicher-Management und Virtualisierung. Die Studierenden können die kennengelernten Ansätze hinsichtlich ihrer Eignung für einen bestimmten Einsatzzweck beurteilen. Der praktische Teil befasst sich mit dem Transfer und der Anwendung des im theoretischen Teil vermittelten Wissens. Sie benutzen Programme und Programmierschnittstellen aus dem Unix-Umfeld, um ein erarbeitetes Lösungskonzept mit dem Schwerpunkt Ein-/Ausgabe durch einen Programmcode zu realisieren.</p> <p>Schlüsselqualifikationen: Qualitätsbewusstsein, Akribie; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Kenntnisse der Vor-/Nachteile von Entwurfsalternativen, Bewertung im jeweiligen Anwendungszusammenhang; Fachspezifische Vertiefungen; Umsetzen fachlicher Lösungskonzepte in Programme und Modelle.</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 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) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Weiterführende Betriebssystemkonzepte (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		

Inhalte:

Die Vorlesung beinhaltet u. a. folgende Themen:

- Grundlagen eines Betriebssystems
 - Aufgaben
 - Struktur und Aufbau
 - Historie
 - Shell
- Prozesse, Threads und Interrupts
 - Prozesse und Threads
 - HW-Interrupts
- Scheduling
 - Grundlagen des Scheduling
 - Echtzeit-Scheduling
- Speicher
 - Interaktion
 - Speicherkonzepte
 - Dateisysteme
- Virtualisierung

Literatur:

- Folien
- Andrew S. Tanenbaum: "Moderne Betriebssysteme", Pearson Studium, ISBN:978-3-8273-7342-7

Modulteil: Weiterführende Betriebssystemkonzepte (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Weiterführende Betriebssystemkonzepte

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

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

Modul INF-0077: Suchmaschinen		8 ECTS/LP
Version 1.2.0 (seit SoSe14) Modulverantwortliche/r: PD Dr. Markus Endres		
<p>Lernziele/Kompetenzen: 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. Hierfür sind vertiefende Kenntnisse im Bereich der Suchmaschinentechologie notwendig. Darunter zählen z.B. die Architektur einer Suchmaschine, Web Crawler, Textprocessing, Indexierung, verschiedene Retrieval Modelle, die Evaluation von Suchalgorithmen, Clusterverfahren, Link Analyse, uvm.</p> <p>Die Studierenden erwerben somit Kenntnisse verschiedener Suchmaschinenaspekte und können die Vor- und Nachteile in praxisrelevanten Aufgabenstellungen abwägen und beurteilen. Zudem können die Studierenden nach der Veranstaltung wissenschaftlich aussagekräftige Bewertungen erstellen und diese in praxisrelevante Lösungsstrategien umsetzen. In der Veranstaltung werden zudem relevante Themen aus der Forschung anhand aktueller Veröffentlichungen angesprochen, so dass die Studierenden die Fähigkeit erwerben, ihr vorhandenes Wissen selbständig zu erweitern.</p>		
<p>Schlüsselqualifikationen: Fachspezifische Vertiefung; Umsetzen fachlicher Lösungskonzepte in Programme und Modelle; Kenntnisse der Vor-/Nachteile von Entwurfsalternativen und Bewertung im jeweiligen Anwendungszusammenhang; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit, mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen anzustellen; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen;</p>		
<p>Arbeitsaufwand: 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>Voraussetzungen: Modul Datenbanksysteme (INF-0073) - empfohlen</p>		
Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Suchmaschinen (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4</p>		
<p>Inhalte: Die Vorlesung behandelt grundlegende Konzepte von Suchmaschinen, Volltext-Suche, SQL-Suchmaschinen und Präferenz-Suchmaschinen (Preference SQL) sowie deren Implementierung.</p>		

Literatur:

- M. Levene: An Introduction to Search Engines and Web Navigation
- R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval
- I. H. Witten, M. Gori, T. Numerico: Web Dragons
- W. Kießling: Foundations of Preferences in Database Systems
- W. Kießling: Preference Queries with SV-Semantics

Modulteil: Suchmaschinen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Suchmaschinen (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0078: Datenbankprogrammierung (Oracle)		5 ECTS/LP
Version 1.2.0 (seit SoSe14) Modulverantwortliche/r: PD Dr. Markus Endres		
<p>Lernziele/Kompetenzen: Nach der Teilnahme an der Veranstaltung sind die Studierenden in der Lage vertiefte Oracle-Kenntnisse in praxisrelevanten Aufgaben anzuwenden. Hierzu ist erweitertes Wissen im Bereich DB-System-Architektur, DB-Entwurfstheorie, Administration, erweitertes SQL, PL/SQL-Programmierung, Baumstrukturen, XML und OLAP notwendig. Die Studierenden erlangen eine fachspezifische Vertiefung im Bereich der Datenbanksysteme und somit die Fertigkeit zur Analyse und Strukturierung von Problemstellungen. Darüber hinaus können die Studierenden diese komplexen, praxisrelevanten Aufgabenstellungen 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 unter praxisnahen Randbedingungen verfassen.</p> <p>Schlüsselqualifikationen: Umsetzen fachlicher Lösungskonzepte in Programm eund Modelle, Fachspezifische Vertiefung; Fachübergreifende Kenntnisse; Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen; Kenntnisse der Vor-/Nachteile von Entwurfsalternativen und Bewertung im jeweiligen Zusammenhang; Auswahl und sichere Anwendung geeigneter Methoden; Kenntnisse von praxisrelevanten Aufgabenstellungen; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen;</p>		
<p>Arbeitsaufwand: 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. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)</p>		
Voraussetzungen: Modul Datenbanksysteme (INF-0073) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Datenbankprogrammierung (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: 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.		
Literatur: <ul style="list-style-type: none"> • R. Elmasri, S. Navathe: Fundamentals of Database Systems • S. Melton: Understanding the New SQL: A Complete Guide • Oracle 11g Online-Dokumentation 		

Zugeordnete Lehrveranstaltungen:

Datenbankprogrammierung (Oracle) (Vorlesung)

Datenbanken haben sich als allgegenwärtiges Werkzeug im öffentlichen, wissenschaftlichen und wirtschaftlichen Leben etabliert. Diese Vorlesung wendet sich an DB-Interessierte, die ihre vorhandenen Kenntnisse aus einer grundlegenden Datenbankvorlesung mit Hilfe von Oracle vertiefen bzw. erweitern wollen. Daher ist die Vorlesung insbesondere für Studierende geeignet, die ihren Schwerpunkt im Bereich Datenbanken- und Informationssysteme setzen bzw. vertiefte praktische Kenntnisse erwerben wollen.

Modulteil: Datenbankprogrammierung (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Datenbankprogrammierung (Oracle) (Übung)

Prüfung

Datenbankprogrammierung (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

Modul INF-0088: Bayesian Networks		5 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: Bayessche Netze gehören zu den vielseitigsten statistischen Methoden des maschinellen Lernens. Nach einer erfolgreichen Teilnahme an diesem Modul verstehen und vertiefen Teilnehmer die Kernprinzipien von Bayesschen Netzen und können diese auf viele praxisrelevante Probleme in unterschiedlichen Fachgebieten anwenden. Diese umfassen unter anderem die Robotik, Websuche, intelligente Agenten, automatisierte Diagnosesysteme und medizinische Systeme. Die Studierenden sind in der Lage, Bayessche Netze zu verstehen, anzuwenden und fachübergreifende Problemstellungen in diesem Kontext zu analysieren und zu evaluieren. Die Teilnahme an diesem Modul fördert Fertigkeiten zum logischen, analytischen und konzeptionellen Denken. Studierende sind in der Lage, mittels Bayesscher Netze wissenschaftlich aussagekräftige Bewertungen zu erstellen.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Logik; umsetzen fachlicher Lösungskonzepte in Modelle; fachübergreifende Kenntnisse; Entwicklung und Umsetzung von Lösungsstrategien komplexer Problemstellungen; systematische Weiterentwicklung von Entwurfsmethoden; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen</p>		
<p>Bemerkung: Die gleichzeitige Einbringung von diesem Modul und INF-0263 ist nicht möglich.</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 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) 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)</p>		
<p>Voraussetzungen: keine</p>		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 5.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Bayesian Networks (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		
<p>Inhalte:</p> <ol style="list-style-type: none"> 1. Basics of Probability Theory 2. Example: Bayesian Network based Face Detection 3. Inference 4. Influence Diagrams 5. Parameter Learning 6. Example: probabilistic Latent Semantic Analysis (pLSA) 		

Literatur:

- Richard E. Neapolitan. Learning Bayesian Networks. Prentice Hall Series in Artificial Intelligence, 2004. ISBN 0-13-012534-2
- Daphne Koller, Nir Friedman. Probabilistic Graphical Models: Principles and Techniques. The MIT Press, 2009. 978-0262013192

Modulteil: Bayesian Networks (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Bayesian Networks (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

Modul INF-0092: Multimedia II: Machine Learning and Computer Vision		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul besitzen die Studierenden fortgeschrittene Kenntnisse des maschinellen Lernens (Entscheidungsbäume, neuronale Netze und tiefe neuronale Netze, Hypothesenevaluation, Instanz-basiertes Lernen, Bayessches Lernen, Lerntheorie), der Datenreduktion (z.B. Hauptkomponentenanalyse), der fortgeschrittenen Bildverarbeitung und des maschinellen Sehens und können diese anwenden. Sie sind in der Lage, wissenschaftlich komplexe Verfahren auf dem Gebiet der multimedialen Datenverarbeitung zu analysieren, zu verstehen und programmatisch umzusetzen, sowie die erlernten Prinzipien auf neue Fragestellungen geeignet anzuwenden. Sie entwickeln Fertigkeiten zur logischen, analytischen und konzeptionellen Denken im Bereich der digitalen Signalverarbeitung und multimedialen Datenverarbeitung.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Logik; umsetzen fachlicher Lösungskonzepte in Modelle; fachübergreifende Kenntnisse; Entwicklung und Umsetzung von Lösungsstrategien komplexer Problemstellungen; systematische Weiterentwicklung von Entwurfsmethoden; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 60 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 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)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Multimedia II: Machine Learning and Computer Vision (Vorlesung)</p> <p>Lehrformen: Vorlesung Sprache: Deutsch SWS: 4</p> <p>Inhalte: Die Vorlesung gibt einen guten Überblick über alle Aspekte des maschinellen Lernens und der maschinellen Extraktion von Informationen aus Multimediadaten (z.B. "GoogleImage 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)</p>

Literatur:

Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.

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

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Multimedia II: Machine Learning and Computer Vision (Klausur)

Klausur / Prüfungsdauer: 120 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

Modul INF-0093: Probabilistic Robotics		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: Nach einer erfolgreichen Teilnahme an diesem Modul verstehen Teilnehmer Grundlagen und vertiefende Fragestellungen und Algorithmen der Robotik (z.B. rekursive Zustandsschätzung, gaußsche- und nicht-parametrische Filter, Kalman Filter, Bewegung und Lokalisierung, Perzeption, Kartierung, SLAM) aus wahrscheinlichkeitstheoretischer Sicht und können erlernte Konzepte auf komplexe, praxisrelevante Aufgabenstellung anwenden. Die Studierenden sind in der Lage, Problemstellungen in diesem Kontext zu analysieren und zu evaluieren. Die Teilnahme an diesem Modul fördert Fertigkeiten zum logischen, analytischen und konzeptionellen Denken auf dem Gebiet der wahrscheinlichkeitstheoretischen Robotik. Studierende können aus den erlernten Konzepten zielgerichtet geeignete Methoden auswählen, sicher anwenden und auf neue, auch fachfremde, Fragestellungen übertragen. Das Modul vermittelt Kompetenzen zum Erkennen aktueller Forschung und bedeutenden technischen Entwicklungen auf diesem Gebiet.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Logik; umsetzen fachlicher Lösungskonzepte in Modelle; fachübergreifende Kenntnisse; Entwicklung und Umsetzung von Lösungsstrategien komplexer Problemstellungen; systematische Weiterentwicklung von Entwurfsmethoden; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Probabilistic Robotics (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte:</p> <ol style="list-style-type: none"> 1. Introduction to Probabilistic Robotics 2. Recursive State Estimation 3. Gaussian Filters 4. Nonparametric Filters 5. Robot Motion 6. Robot Perception 7. Mobile Robot Localization: Markow and Gaussian 8. Mobile Robot Localization: Grid and MonteCarlo 9. Occupancy Grid Mapping 10. SLAM 		

Literatur:

Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic Robotics. Springer Verlag.

Zugeordnete Lehrveranstaltungen:

Probabilistic Robotics (Vorlesung)

In the course of this lecture students will learn how robots can estimate their state (e.g. their pose) in a probabilistic fashion, i.e. in the face of uncertainty. The main focus of this lecture is on the Bayes Filter algorithm which enables robots to estimate their new state after executing a control and to incorporate sensor measurements to update their belief. Various flavors of the Bayes Filter such as the Kalman Filter and the Particle Filter will be discussed in detail. Furthermore, students will get to know different ways to model robot motion and measurements of various types of sensors. The final chapters of the lecture will be on approaches to robot localization, i.e. the problem of the robot having to determine its position on a given map of the environment. Also, the localization problem will be discussed for situations when the robot has to generate a map itself by occupancy grid mapping or simultaneous localization and mapping (SLAM) algorithms.

... (weiter siehe Digicampus)

Modulteil: Probabilistic Robotics (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Probabilistic Robotics (Übung)

Prüfung

Probabilistic Robotics (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

Modul INF-0112: Graphikprogrammierung		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
Lernziele/Kompetenzen: Die Studierenden beherrschen die 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. Schlüsselqualifikationen: 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		
Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Übung (Präsenzstudium) 60 Std. Vorlesung (Präsenzstudium) 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)		
Voraussetzungen: Mathematik für Informatiker I + II (alternativ Analysis I + Lineare Algebra I) empfohlen Modul Informatik 1 (INF-0097) - empfohlen Modul Informatik 2 (INF-0098) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 4.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
Modulteil: Graphikprogrammierung (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4
Inhalte: Koordinaten und Transformationen, Projektionen und Kameramodelle, Sichtbarkeit, Farbmodelle, Beleuchtung und Schattierung, Texturen, Schattenberechnung, Raytracing, Animationstechniken, OpenGL/JOGL
Literatur: <ul style="list-style-type: none"> • Eigenes Skriptum • M. Bender, M. Brill, Computergrafik - ein anwendungsorientiertes Lehrbuch, Hanser 2006 • F. Hill, S. Kelley: Computer graphics using OpenGL, Pearson 2007
Modulteil: Graphikprogrammierung (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 2

Prüfung

Graphikprogrammierung (Klausur)

Klausur / Prüfungsdauer: 120 Minuten

Modul INF-0116: Algebraische Semantik und Algebraische Systementwicklung		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p>Lernziele/Kompetenzen: 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>Schlüsselqualifikationen: analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis</p>		
<p>Arbeitsaufwand: 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 anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p>		
Voraussetzungen: Modul Diskrete Strukturen für Informatiker (INF-0109) - empfohlen		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Algebraische Semantik und Algebraische Systementwicklung (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 4		
Inhalte: Halbringe, Testelemente, Modale Operatoren, Iterationsoperatoren, Terminierungsanalyse, Wissens-/ Glaubenslogiken, Temporale Logiken, Algebra paralleler Systeme.		
Literatur: <ul style="list-style-type: none"> • Eigenes Skriptum • U. Hebisch, H. J. Weinert: Halbringe - Algebraische Theorie und Anwendungen in der Informatik, Teubner 1993 		
Modulteil: Algebraische Semantik und Algebraische Systementwicklung (Übung)		
Lehrformen: Übung		
Sprache: Deutsch		
SWS: 2		
Prüfung		
Algebraische Semantik und Algebraische Systementwicklung (Klausur) Klausur / Prüfungsdauer: 120 Minuten		

Modul INF-0117: Funktionale Modellierung für Geoinformationssysteme		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller Prof. Dr. Sabine Timpf		
<p>Lernziele/Kompetenzen: Die Studierenden verfügen über ein tiefes 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>Schlüsselqualifikationen: analytisch-methodische Kompetenz; Abwägen von Lösungsansätzen; Abstraktionsfähigkeit; Training des logischen Denkens; eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Grundsätze guter wissenschaftlicher Praxis</p>		
<p>Arbeitsaufwand: 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)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Funktionale Modellierung für Geoinformationssysteme (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: 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		
Literatur: <ul style="list-style-type: none"> • Eigenes Skriptum • B O'Sullivan, D. Stewart, J. Goerzen: Real World Haskell, O'Reilly 2008 • M.Worboys, M. Duckham: GIS - A computing perspective, Routledge 2004 		
Modulteil: Funktionale Modellierung für Geoinformationssysteme (Übung)		
Lehrformen: Übung		
Sprache: Deutsch		
SWS: 2		

Prüfung

Funktionale Modellierung für Geoinformationssysteme (Klausur)

Klausur / Prüfungsdauer: 120 Minuten

Modul INF-0129: Softwaretechnik II		8 ECTS/LP
Version 2.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen:</p> <p>Fachbezogene Kompetenzen:</p> <p>Nach dem erfolgreichen Besuch des Moduls sind die Teilnehmer in der Lage, fortgeschrittene Verfahren der agilen Softwareentwicklung, des Requirements Engineerings, des Testens und des Refactorings von Softwaresystemen anzuwenden und miteinander zu verbinden. Sie können die Anwendbarkeit der verschiedenen Verfahren und Methoden situationsspezifisch beurteilen.</p> <p>Die Teilnehmer verstehen die Aufgaben des Requirements Engineering und sind in der Lage, wesentliche Methoden der Requirements-Erfassung und Dokumentation (Use-Cases, Satzschablonen, formale Modelle) anzuwenden und die Eignung der verschiedenen Dokumentationsformen im Projektkontext zu bewerten. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken und können so systematisch komplexe Kundenanforderungen analysieren und können diese formulieren und in geeigneten Modellen darstellen.</p> <p>Die Studierenden verstehen einen agilen Entwicklungsprozess und die Basisprinzipien agiler Vorgehensweisen. Sie können den agilen Entwicklungsprozess auf Projekte übertragen und seine Anwendbarkeit im Projektkontext beurteilen. Sie kennen agile Praktiken und deren Wechselwirkungen und können die Praktiken einsetzen. Sie verstehen die Aufgaben der beteiligten Rollen und die Bausteine des Entwicklungsprozesses.</p> <p>Die Studierenden verstehen die Aufgaben des Testens und die Teilschritte des Testprozesses. Sie kennen Verfahren zur Bestimmung von Testfällen und zur Bewertung der Adäquatheit von Testsuiten und können diese anwenden.</p> <p>Die Studierenden 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>Die Studierenden erwerben insgesamt Kenntnisse in Methoden zur Entwicklung großer Softwaresysteme und der Konstruktion von Abstraktionen sowie die Fähigkeit zu deren Anwendung im Projektkontext. Sie können alternative Vorgehensweisen wissenschaftlich fundiert bewerten und für ein Projekt geeignete auswählen.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Abstraktionsfähigkeit • Moderieren fachlicher Sitzungen • Erlernen des eigenständigen Arbeitens mit Lehrbüchern und Arbeit in selbstorganisierten Teams • Kenntnisse von praxisrelevanten Aufgabenstellungen 		
<p>Arbeitsaufwand:</p> <p>Gesamt: 240 Std.</p> <p>90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> <p>30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> <p>60 Std. Vorlesung (Präsenzstudium)</p>		
<p>Voraussetzungen:</p> <p>Programmierkenntnisse in Java (empfohlen)</p> <p>Modul Softwaretechnik (INF-0120) - empfohlen</p>		
<p>Angebotshäufigkeit: jedes Wintersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>

SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Moduleile		
Modulteil: Softwaretechnik II (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte:		
<p>Agile Softwareentwicklung:</p> <ul style="list-style-type: none"> • Entwicklungsmethoden (Scrum) • Agile Praktiken • Agile Werte, Prinzipien und Methoden <p>Refactoring</p> <ul style="list-style-type: none"> • Code Smells • Prinzipien des objektorientierten Designs • Wichtige Refactorings <p>Testen</p> <ul style="list-style-type: none"> • Testprozess und Ziele des Testens • Testarten • Methoden zur Testfallgewinnung • Adäquatheitskriterien beim Blackbox- und Whitebox-Testen <p>Requirements Engineering</p> <ul style="list-style-type: none"> • Aufgaben, Begriffe und Artefakte • RE-Prozess • Techniken zur Requirements-Elicitation, -Analyse und -Dokumentation • Qualitätskriterien für Software-Requirements 		
Literatur:		
<ul style="list-style-type: none"> • 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 		
Zugeordnete Lehrveranstaltungen:		
Softwaretechnik 2 (Vorlesung)		
Modulteil: Softwaretechnik II (Übung)		
Lehrformen: Übung		
Sprache: Deutsch		
SWS: 4		
Zugeordnete Lehrveranstaltungen:		
Übung zu Softwaretechnik 2 (Übung)		

Prüfung

Softwaretechnik II Klausur

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0130: Formale Methoden im Software Engineering		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
Lernziele/Kompetenzen: Fachbezogene Kompetenzen: Die Studierenden können mathematisch-formale Methoden für die Programmverifikation, speziell bei sicherheitskritischer Software einsetzen. Durch das Verständnis von Kalkülen trainieren sie die Fertigkeit zum logischen, konzeptuellen und analytischen Denken. Sie können Spezifikationen von Datenstrukturen entwerfen und deren Eigenschaften formal beweisen. Sie sind in der Lage, funktionale Eigenschaften von sequentiellen und nebenläufigen Programmen zu formulieren und dafür Beweise zu erstellen. Sie erwerben die Fertigkeit zur abstrakten, modularen Modellierung von Softwaresystemen und zur systematischen Entwicklung von korrekten Programmen aus diesen Modellen. Sie erwerben dadurch die Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen und können mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen abgeben. Sie erwerben Fähigkeiten, um Beiträge zur Wissenschaft zu leisten.		
Schlüsselqualifikationen: <ul style="list-style-type: none"> • Training des logischen Denkens • Analytisch-methodische Kompetenz • Erwerb von Abstraktionsfähigkeiten • Fähigkeiten zur Modellbildung für Softwaresysteme 		
Arbeitsaufwand: 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)		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Formale Methoden im Software Engineering (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2		
Inhalte: 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		

Literatur:

- Sperschneider, Antoniou: Logic: A Foundation for Computer Science, Addison Wesley 1991
- Loeckx, Ehrich, Wolf: Specification of Abstract Data Types, Wiley 1996
- Ausführliche Dokumentation
- Folienhandout

Modulteil: Formale Methoden im Software Engineering (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Formale Methoden im Software Engineering (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Modul INF-0131: Software- und Systemsicherheit		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen:</p> <p>Fachbezogene Kompetenzen:</p> <p>Nach der erfolgreichen Teilnahme an der Lehrveranstaltung kennen Studierende die Grundlagen, Aufbau, und typische Schwächen kryptographischer Protokolle und praxisrelevanter, sicherheitskritischer Systeme, können diese verständlich und sicher darstellen, und verstehen die Grundprinzipien deren Sicherheitsanalyse.</p> <p>Sie können Bedrohungsanalyse durchführen, kryptographische Protokolle entwickeln, Chipkarten programmieren, sicherheitskritische Systeme in Teams entwerfen, und ihre Ergebnisse dokumentieren.</p> <p>Sie können systematisch Bedrohungen für Softwaresysteme analysieren und deren Risiken bewerten.</p> <p>Sie können einen modellgetriebenen Entwicklungsprozess für sicherheitskritische Systeme anwenden.</p> <p>Sie sind in der Lage, bedeutende technische Entwicklungen zu erkennen und fachliche Lösungskonzepte in Programme umzusetzen. Sie haben Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Abstraktionsfähigkeiten • Grundsätze guter wissenschaftlicher Praxis • Mathematisch-formale Grundlagen • Quantitative Aspekte der Informatik • Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen • Fähigkeit, vorhandenes Wissen selbstständig zu erweitern • Qualitätsbewusstsein und Akribie 		
<p>Arbeitsaufwand:</p> <p>Gesamt: 240 Std.</p> <p>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>60 Std. Übung (Präsenzstudium)</p>		
Voraussetzungen:		
keine		
Angebotshäufigkeit: unregelmäßig (i. d. R. im WS)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Software- und Systemsicherheit (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p>		

Inhalte:

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.

Literatur:

- Schneier: Applied Cryptography, Wiley and Sons, 1996 (2nd edition)
- Anderson, Needham: Programming Satan's Computer, in: Computer Science Today, Springer LNCS 1000, 1995
- 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
- Folienhandout, Spezifikationen und APIs

Modulteil: Software- und Systemsicherheit (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Software- und Systemsicherheit (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Modul INF-0133: Selbstorganisierende, adaptive Systeme		8 ECTS/LP
Version 1.0.0 (seit SoSe14 bis SoSe18) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen: Die Studierenden haben vertiefte Fachkenntnisse über 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. Die Studierenden sind in der Lage, adaptive Systeme adäquat zu modellieren und dokumentieren.</p> <p>Außerdem kennen die Studierenden praxisrelevante Fragestellungen.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten • Grundsätze guter wissenschaftlicher Praxis 		
<p>Arbeitsaufwand: Gesamt: 240 Std. 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) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Selbstorganisierende, adaptive Systeme (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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>		

Literatur:

- Gleick: Chaos: Making a New Science, Penguin 2008
- Strogatz: Sync : The Emerging Science of Spontaneous Order, Hyperion 2003
- Miller, Page: Complex Adaptive Systems: An Introduction to Computational Models of Social Life, Princeton University Press 2007
- Dawkins: The Selfish Gene, Oxford University Press, 3rd Revised Edition
- Wolfram: A New Kind of Science, Wolfram Media Inc. 2002
- von Neumann, Morgenstern: Theory of Games and Economic Behavior, 2004
- Folienhandout

Zugeordnete Lehrveranstaltungen:

Selbst-organisierende, adaptive Systeme (Vorlesung)

"Selbst-organisierende, adaptive Systeme" beschäftigt sich mit theoretischen Grundlagen, die für die Entwicklung von offenen Multiagentensystemen nötig sind. Dabei folgt die Vorlesung vor allem dem Aufbau von <http://www.masfoundations.org/> und behandelt Spieltheorie, Mechanism Design und (verteilte) Constraint-Optimierung. Sie richtet sich vor allem an den Anwendungsfällen des Lehrstuhls "Verteiltes Energiemanagement" (<http://www.focas-reading-room.eu/robust-optimization-in-self-organizing-power-management/>) sowie flexible Produktion (<https://www.isse.uni-augsburg.de/projects/reif/completed/save-orca/>) und Multi-Roboter Arbeitszuweisung im Ensemble Programming (<https://www.isse.uni-augsburg.de/projects/reif/robot-ensembles/>) aus.

Modulteil: Selbstorganisierende, adaptive Systeme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Selbst-organisierende, adaptive Systeme (Übung)

Prüfung

Selbstorganisierende, adaptive Systeme (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Modul INF-0145: Mikrorechnertechnik und Echtzeitsysteme		6 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen in den folgenden Bereichen auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: Grundlagen und Aufbau von Mikrorechnern und Mikrocontrollern, Echtzeitprogrammierung und -betriebssysteme, funktionale Sicherheit, Automotive- und Avionic-Systeme. Sie verstehen die Konzepte hinter und den Aufbau von Mikrocontrollern und können einordnen, wo diese eingesetzt werden. Sie lernen Kriterien, anhand derer sie auswählen können, welcher Mikrocontroller für welchen Zweck sinnvoll eingesetzt werden kann. Dazu können die Studierenden beispielsweise deren Energieeffizienz einschätzen. Um die Erkenntnisse direkt in die Praxis zu übertragen, analysieren die Studierenden am Beispiel einer Wetterstation, wie ein konkretes System realisiert werden kann.</p> <p>Sie können die verschiedenen Arten von Echtzeitsystemen differenzieren und entscheiden, welche Echtzeit-Schedules für ein bestimmtes Echtzeitsystem geeignet sind. Weiterhin lernen die Studierenden verschiedene Techniken der Echtzeitprogrammierung, was auch die Auswahl der zu einer Anwendung passenden Ablaufsteuerung miteinschließt. Schließlich werden noch Standards der funktionalen Sicherheit (SIL, ASIL) bewertet.</p> <p>Schlüsselqualifikationen: Analytisch-methodische Kompetenz, Abwägung von Lösungsansätzen, Präsentation von Lösungen von Übungsaufgaben; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein, Akribie; Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 15 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium) 75 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Mikrorechnertechnik und Echtzeitsysteme (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 3</p>		

Inhalte:

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.

Literatur:

- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010
- Heinz Wörn, Uwe Brinkschulte, Echtzeitsysteme, Springer Verlag, Berlin/Heidelberg, 2005

Zugeordnete Lehrveranstaltungen:**Mikrorechnertechnik und Echtzeitsysteme (Vorlesung)**

Die MA-Vorlesung "Mikrorechnertechnik und Echtzeitsysteme" behandelt aufbauend auf der BA-Vorlesung "Systemnahe Informatik" die grundlegenden Prinzipien der Mikrocontroller und der eingebetteten Systeme. In der Praxis häufig verwendete Mikrocontroller und Bustechnologien werden in ihrer Funktionsweise analysiert und zukunftsweisende Technologien dieser Bausteine erläutert. Weiterhin wird auf Anforderungen eingebetteter Echtzeitsysteme eingegangen. Echtzeitanwendungen finden sich im Flugzeugbau, in der Motorsteuerung und in Fahrerassistenzsystemen in Autos, in der Kraftwerkssteuerung und in vielen industriellen Maschinen. Für solche Anwendungen werden die Grundlagen der Echtzeitsysteme bereitgestellt.

Modulteil: Mikrorechnertechnik und Echtzeitsysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 1

Zugeordnete Lehrveranstaltungen:

Übung zu Mikrorechnertechnik und Echtzeitsysteme (Übung)

Prüfung**Mikrorechnertechnik und Echtzeitsysteme (Klausur)**

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0147: Prozessorarchitektur		5 ECTS/LP
Version 1.1.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden verstehen die Prinzipien des Aufbaus folgender Prozessoren auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: superskalare Mikroprozessoren, Multi- und Manycore-Prozessoren, Signalprozessoren, Smartphone-Prozessoren und GPUs. Außerdem lernen sie aktuelle Konzepte der Prozessorarchitektur kennen und können Vor-/Nachteile aktueller und zukünftiger Prozessoren anhand ihres internen Aufbaus einschätzen und beurteilen. Sie verstehen die Funktionsweise von verschiedenen Komponenten von Mikroprozessoren und deren Zusammenspiel. Somit ist es den Studierenden möglich, den Einfluss verschiedener Architekturerweiterungen auf das Gesamtsystem einzuordnen. Ebenso können sie unterscheiden, wann es sinnvoll ist, speichergekoppelte, nachrichtengekoppelte oder datenparallele Prozessoren einzusetzen, und differenzieren zwischen Techniken für energieeffiziente oder hochperformante Prozessoren. In der Übung simulieren die Studierenden verschiedene Prozessorarchitekturen und Sprungvorhersagetechniken und bewerten deren Einfluss auf die Laufzeit und den Energieverbrauch.</p> <p>Schlüsselqualifikationen: Analytisch-methodische Kompetenz, Abwägung von Lösungsansätzen, Präsentation von Lösungen von Übungsaufgaben; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein, Akribie; Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen; Fertigkeit, mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen anzustellen; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen</p>		
<p>Arbeitsaufwand: 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>		
<p>Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen</p>		
Angebotshäufigkeit: unregelmäßig (i. d. R. im WS)	Empfohlenes Fachsemester: 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Prozessorarchitektur (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: Die Vorlesung "Prozessorarchitektur" vertieft die Techniken superskalarer Mikroprozessoren und aktueller Multicore-Prozessoren. Dabei werden die Pipelinestufen detailliert behandelt, mehrfädige Prozessoren und Multicores gegenübergestellt sowie aktuelle Beispielprozessoren vorgestellt. Außerdem wird aus der Forschung an Manycores und Echtzeit-Multicores berichtet.</p>		

Literatur:

- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010
- John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011

Zugeordnete Lehrveranstaltungen:

Prozessorarchitektur (Vorlesung)

Ausgehend von den Herausforderungen bei der Leistungssteigerung moderner Prozessoren werden in der Vorlesung bereits realisierte Konzepte wie Superskalartechnik, Mehrfädigkeit und Out-of-Order-Ausführung behandelt. Darauf aufbauend werden weitere Architekturtypen wie GPUs vorgestellt sowie aktuelle Forschungsthemen, z.B. Manycore-Prozessoren, betrachtet.

Modulteil: Prozessorarchitektur (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Prozessorarchitektur (Übung)

Prüfung

Prozessorarchitektur (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

Modul INF-0148: Entwurf und Analyse fehlertolerierender Rechensysteme		5 ECTS/LP
Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden verstehen die folgenden Konzepte auf einem fortgeschrittenen, wissenschaftlichen Niveau: Ursachen, Effekte und Modellierung von Hardware-Fehlern, stochastische Grundlagen der Fehlerrechnung, Modellierung und Analyse von Rechensystemen mit Zuverlässigkeitsblockdiagrammen, Fehlerbäumen und FMEA, unterschiedliche Redundanzarten und ihre Verwendung in verschiedenen Hard- und Software-Techniken zur Fehlererkennung, Fehlerkorrektur, Fehlerdiagnose und Rekonfiguration. Weitere Schwerpunkte sind fehlertolerierende Speicherstrukturen und Fehlertoleranzkonzepte in sicherheitskritischen Systemen am Beispiel der ISO 26262 Norm.</p> <p>Die Studierenden können die Ursachen und Effekte von Hardware-Fehlern nachvollziehen und sind in der Lage unterschiedliche Redundanztechniken analysieren und bewerten zu können. Sie lernen verschiedene Konzepte fehlertolerierender Rechensysteme aus Theorie und Praxis, wie z.B. Lockstep-Ausführung, Prozess-Redundanz, ECC-Speicher-Implementierungen oder Kontrollflussüberprüfung, kennen, und können deren jeweilige Vor- und Nachteile einschätzen. In den praktischen Übungen wenden die Studierenden die Methoden der Modellierung und Fehlerrechnung an verschiedenen Aufgaben an, präsentieren ihre Lösungen und vergleichen auf qualitativer Ebene verschiedene Fehlertoleranzkonzepte aus Forschung und Praxis.</p> <p>Schlüsselqualifikationen: Analytisch-methodische Kompetenz in der Analyse fehlertolerierender Rechensysteme, eigenständiges Arbeiten mit wissenschaftlichen Aufsätzen, Abwägung von Lösungsansätzen, Verständliche Präsentation von Übungsaufgaben, Selbstreflexion</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 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) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Entwurf und Analyse fehlertolerierender Rechensysteme (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		

Inhalte:

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.

Literatur:

- D.P. Siewiorek, R.S. Swarz: Reliable Computer Systems, Peters, 1998
- I. Koren, C.M. Krishna: Fault Tolerant Systems, Morgan Kaufmann, 2007
- T. Anderson, P.A. Lee: Fault Tolerance - Principles and Practice, Prentice Hall, 1982

Modulteil: Entwurf und Analyse fehlertolerierender Rechensysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Entwurf und Analyse fehlertolerierender Rechensysteme (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

Modul INF-0149: Praktikum Eingebettete Systeme		5 ECTS/LP
Version 1.1.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen in den folgenden Bereichen auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: selbstständiges Arbeiten mit Mikrocontrollern, Datenblättern und Spezifikationen, Anbindung von analoger und digitaler Peripherie, Entwurf und Modellierung von eingebetteter Software mit Zustandsdiagrammen sowie deren Transformation in Code. Weitere Schwerpunkte sind die Konfiguration von sequentiellen Schnittstellen sowie Scheduling und task-basierte Programmierung.</p> <p>Durch die Implementierung von Aufgaben für einen Mikrocontroller setzen sie die erlernten Konzepte direkt in die Praxis um. Dabei steht Interaktion mit Sensoren und Aktoren sowie die Kommunikation mit anderen Systemteilen im Vordergrund. Dazu können sie unterschiedliche Arten der Ablaufsteuerung unterscheiden und anwenden. In einer abschließenden Projektphase lernen die Studierenden, eine komplexe Aufgabenstellung im Team zu planen, nach einem selbst entwickelten fundierten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zur verständlichen Darstellung und Dokumentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein, Akribie; Projektgebundene Arbeit und Zeitmanagement; Auswahl und sichere Anwendung geeigneter Methoden; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Selbstreflexion</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p>Voraussetzungen: Kenntnisse in C. Modul Mikrorechner- und Echtzeitsysteme (INF-0145) - empfohlen</p>		
<p>Angebotshäufigkeit: jedes Sommersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 4</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Praktikum Eingebettete Systeme Lehrformen: Praktikum Sprache: Deutsch SWS: 4</p>		
<p>Inhalte: 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.</p>		

Literatur:

- Marwedel, Wehmeyer: Eingebettete Systeme, Springer Verlag, Heidelberg, 2007
- Wiegmann: Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller, 6. Auflage, VDE Verlag, Berlin, 2011

Prüfung

Erfolgreiche Praktikumsteilnahme, Projektvorstellung

Praktikum

Modul INF-0150: Hardware-Entwurf		8 ECTS/LP
Version 1.1.0 (seit WS14/15) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen in den folgenden Bereichen auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: Entwurfsprozess für digitale Schaltungen, Schaltungslogik und Gatter, physikalische Grundlagen der elektronischen Bauteile, Beschreibung von Hardware mit einer Hardwarebeschreibungssprache. Zunächst lernen die Studierenden die Verknüpfung logischer Gatter und bauen einen Halb- und einen Volladdierer. Sie verstehen den Entwicklungsprozess für digitale Schaltungen und wenden diesen direkt praktisch an, indem sie ihren eigenen RISC-V-Prozessor entwickeln. Sie modellieren und implementieren ihn selbstständig mit Hilfe der Hardwarebeschreibungssprache VHDL. Dazu lernen sie die Vor- und Nachteile von schematischer und textueller Hardware-Beschreibung kennen und können entscheiden, wann es sinnvoll ist, welche Variante einzusetzen. Darüber hinaus kombinieren sie synchrone und asynchrone Prozesse, um ein gutes Zusammenspiel der Komponenten ihres selbstgebauten Mikroprozessors zu erreichen. Schließlich beurteilen die Studierenden die Effizienz ihrer Implementierung anhand der erzielten Taktfrequenz und des benötigten Hardware-Aufwandes. In einer abschließenden Projektphase lernen sie, eine komplexe Aufgabenstellung zu planen, nach einem selbst entwickelten fundierten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zur verständlichen Darstellung und Dokumentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein, Akribie; Projektgebundene Arbeit und Zeitmanagement; Auswahl und sichere Anwendung geeigneter Methoden; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Selbstreflexion</p>		
<p>Bemerkung: Im WiSe 2018/2019 findet statt diesem Modul das Modul "INF-0297: Praktikum Prozessorbau" statt. Das weitere Angebot dieses Moduls ist derzeit noch nicht abzusehen.</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium)</p>		
<p>Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Hardware-Entwurf (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: 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</p>		

Literatur:

- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010
- John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011

Modulteil: Hardware-Entwurf (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Projektvorstellung und Projektabnahme

Praktikum

Beschreibung:

Erfolgreiche Praktikumsteilnahme, Projektvorstellung am Ende des Semesters

Modul INF-0156: Algebraische Beschreibung paralleler Prozesse		6 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p>Lernziele/Kompetenzen:</p> <p>Die Studierenden kennen die Prozessalgebra CCS und formale Beschreibungen verteilter Systeme in CCS. Sie kennen einen Mechanismus, mit dem man in CCS die operationale Semantik definiert, und können diesen Mechanismus auf konkrete Prozesse anwenden. Sie besitzen die Fähigkeit, verteilte Systeme entsprechend auf eine exakte, algebraische Weise zu modellieren, d.h. einen entsprechenden Prozess zu entwerfen.</p> <p>Sie wissen, welche Anforderungen man in diesem Kontext an Äquivalenzbegriffe stellen muss und lernen verschiedene solche Begriffe kennen. Sie können auf verschiedenen Weisen -- z.B. auch durch algebraische Gesetze -- formal prüfen, ob ein System eine, ebenfalls in CCS geschriebene, Spezifikation erfüllt.</p> <p>Sie können beurteilen, ob ein eine algebraische Gleichung ein gültiges Gesetz ist. Zur Vertiefung der Kenntnisse werden die meisten Resultate der Vorlesung bewiesen.</p> <p>Die Studierenden wissen, wie man CCS und die operationale Semantik in einer spezifischen Weise um Zeit erweitern kann, und sind dadurch in der Lage, auch andere Prozessalgebren mit anderen Semantik-Regeln zu verstehen und zu verwenden.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Methodik; Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit zur Analyse und Strukturierung von Informatikproblemstellungen; Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 180 Std.</p> <p>15 Std. Übung (Präsenzstudium)</p> <p>45 Std. Vorlesung (Präsenzstudium)</p> <p>75 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>22 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>23 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p>		
<p>Voraussetzungen:</p> <p>Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen</p> <p>Modul Logik für Informatiker (INF-0155) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: ab dem 5.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 4</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Algebraische Beschreibung paralleler Prozesse (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch</p> <p>SWS: 3</p> <p>Inhalte:</p> <p>Algebraische Spezifikation verteilter Systeme mittels der Prozessalgebra CCS; operationale Semantik mittels SOS-Regeln; Äquivalenz- bzw. Kongruenzbegriffe (starke und schwache Bisimulation, Beobachtungskongruenz); Nachweis von Kongruenzen mittels Axiomen; Einführung in eine Kombination von Bisimulation und Effizienzvergleich.</p> <p>Die meisten Resultate der Vorlesung werden bewiesen.</p>		

Literatur:

- R. Milner: Communication and Concurrency, Prentice Hall
- L. Aceto, A. Ingolfsdottir, K.G. Larsen, J. Srba: Reactive Systems. Cambridge University Press 2007
- J. Bergstra, A. Ponse, S. Smolka (eds.): Handbook of Process Algebras, Elsevier

Modulteil: Algebraische Beschreibung paralleler Prozesse (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 1

Prüfung

Algebraische Beschreibung paralleler Prozesse (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0157: Endliche Automaten		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p>Lernziele/Kompetenzen: Nach der Teilnahme haben die Studierenden vertiefte Kenntnisse in der Automatentheorie und können sie verständlich erläutern. Sie können deterministische Automaten minimieren und das Verfahren mit guter Effizienz automatisieren. Sie können Schaltkreisverhalten und Mealy-Automaten ineinander übersetzen, und sie können mit geeigneten Ergebnissen reguläre von nicht-regulären Sprachen unterscheiden. Zur Vertiefung der Kenntnisse werden die meisten Resultate der Vorlesung bewiesen. Sie haben anspruchsvolle Modellierungen von Problemen mit endlichen Automaten kennengelernt und können sich in neue Anwendungen der Automatentheorie einarbeiten und ihre Korrektheit beurteilen.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Methodik; Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Auswahl und sichere Anwendung geeigneter Methoden; Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 20 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 37 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 48 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 45 Std. Vorlesung (Präsenzstudium)</p>		
<p>Voraussetzungen: Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen Modul Informatik 3 (INF-0111) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: ab dem 5.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 3</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Endliche Automaten (Vorlesung mit integrierter Übung) Lehrformen: Vorlesung + Übung Sprache: Deutsch SWS: 3</p>		
<p>Inhalte: Die Vorlesung vertieft die Kenntnisse über Endliche Automaten aus der Grundvorlesung "Einführung in die theoretische Informatik", insbesondere über Minimierung und Abschlusseigenschaften. Sie behandelt eine Anwendung bei der Lösung diophantischer Gleichungen. Sie stellt Mealy-, Moore- und Büchi-Automaten vor und behandelt die Übersetzungen zwischen Mealy-Automaten und Schaltkreisverhalten. Die meisten Resultate der Vorlesung werden bewiesen.</p>		
<p>Literatur:</p> <ul style="list-style-type: none"> • Hopcroft, (Motwani, Ullman: Introduction to Automata Theory, Languages and Computation; deutsch: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie • Schönig: Theoretische Informatik kurz gefasst. 5. Auflage • Thomas: Automata on Infinite Objects. Chapter 4 in Handbook of Theoretical Computer Science, Hrsg. van Leeuwen 		
<p>Prüfung Endliche Automaten (mündliche Prüfung) Mündliche Prüfung / Prüfungsdauer: 30 Minuten</p>		

Modul INF-0161: Petrinetze - eine Theorie paralleler Systeme		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p>Lernziele/Kompetenzen: Die Studierenden kennen verschiedene Varianten von Petrinetzen und entsprechende formale, Graphen-basierte Beschreibungen paralleler bzw. nebenläufiger Systeme. Anhand verschiedener Verhaltensbegriffe (Schalt- und Schrittfolgen, Sprache, Failure-Semantik) lernen sie die neuartigen Aspekte der Abläufe solcher Systeme kennen. Sie werden befähigt, wichtige Systemeigenschaften mit Petrinetz-spezifischen Methoden zu beweisen oder zu widerlegen. Auf dieser Grundlage können sie parallele bzw. nebenläufige Systeme mit Petrinetzen formal modellieren sowie analysieren, unter welchen Aspekten sich verschiedene Modelle gleichen oder unterscheiden. Sie lernen einige Entscheidbarkeitsprobleme kennen und können eigene Beweise dazu führen. Zur Vertiefung der Kenntnisse werden die meisten Resultate der Vorlesung bewiesen.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Methodik; Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fertigkeit zur Analyse und Strukturierung von Informatikproblemstellungen; Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen		
Angebotshäufigkeit: unregelmäßig (i. d. R. im WS)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Petrinetze - eine Theorie paralleler Systeme (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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); Entscheidbarkeitsprobleme. Die meisten Resultate der Vorlesung werden bewiesen.</p>		
<p>Literatur:</p> <ul style="list-style-type: none"> • Desel, Reisig, Rozenberg (eds.): Lectures on Concurrency and Petri Nets. Advances in Petri Nets. Springer, LNCS 3098 • Peterson: Petri Net Theory and the Modelling of Systems. Prentice Hall • Reisig: Petrinetze - Eine Einführung. 2. Auflage; Springer 		
Zugeordnete Lehrveranstaltungen:		
Petrinetze -- eine Theorie paralleler Systeme (Vorlesung)		

Petrinetze bilden ein formales Hilfsmittel zur Modellierung paralleler Systeme, das durch die graphische Darstellung der Netze besonders attraktiv ist. Die Vorlesung behandelt verschiedene wichtige Klassen von Petrinetzen und zugehörigen Analyseverfahren. Dabei stellt sich die Frage, wie das Verhalten von parallelen Systemen auf einer abstrakten Ebene geeignet zu beschreiben ist; im Rahmen der Theorie der Petrinetze sind hier besonders Beschreibungen wichtig, bei denen ein Ablauf eines Systems durch eine Halbordnung der ausgeführten Aktionen modelliert wird.

Modulteil: Petrinetze - eine Theorie paralleler Systeme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Petrinetze -- eine Theorie paralleler Systeme (Übung)

Prüfung

Petrinetze - eine Theorie paralleler Systeme (Mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0163: Verteilte Algorithmen		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p>Lernziele/Kompetenzen: Die Studierenden lernen formale Modelle zur Beschreibung verteilter Algorithmen in synchronen und asynchronen Netzwerke sowie die entsprechenden Methoden zum Nachweis der Korrektheit und des Aufwands kennen. Sie können beschreiben, wie Fehler in den beiden Modellen behandelt werden. Die Studierenden lernen wichtige Algorithmen und ihren Aufwand kennen, können sie in konkreten Netzwerken simulieren und verstehen ihre Korrektheit. Sie werden befähigt, solche Algorithmen zu modifizieren und zu analysieren, so dass sie zugehörige Korrektheitsbeweise und Aufwandsbestimmungen prüfen oder selbst entwickeln können. Sie erlangen damit ein vertieftes Verständnis für die Probleme und Problemlösungen in verteilten Systemen.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Methodik; 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</p>		
<p>Arbeitsaufwand: 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)</p>		
<p>Voraussetzungen: Modul Einführung in die Theoretische Informatik (INF-0110) - empfohlen Modul Informatik 3 (INF-0111) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	

Modulteile		
Modulteil: Verteilte Algorithmen (Vorlesung)		
<p>Lehrformen: Vorlesung Sprache: Deutsch SWS: 4</p>		
<p>Inhalte: 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 nachgewiesen sowie Korrektheitsbeweise geführt.</p>		
<p>Literatur: Nancy Lynch, Distributed Algorithms. Morgan Kaufmann 1996</p>		
Modulteil: Verteilte Algorithmen (Übung)		
<p>Lehrformen: Übung Sprache: Deutsch SWS: 2</p>		

Prüfung

Verteilte Algorithmen (Mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0175: Multimedia I: Usability Engineering		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an der Lehrveranstaltung haben die Studierenden vertiefte Kenntnisse über aktuelle Paradigmen der (Multimodalen) Mensch-Technik Interaktion und Methoden der Gestaltung und Evaluation von neuartigen multimedialen Interaktionstechniken. Sie sind in der Lage im Team Multimediaprojekte durchzuführen und Ihre Lösungsstrategien kritisch zu evaluieren. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen, Leistung von Beiträgen zur Wissenschaft sowie Kompetenzen in der Evaluation von interaktiven Systemen hinsichtlich traditionellen und modernen Nutzeranforderungen. Außerdem verstehen die Studierenden praxisrelevante Fragestellungen.</p> <p>Schlüsselqualifikationen: Fertigkeit der Zusammenarbeit in Teams; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, Beiträge zur Wissenschaft zu leisten; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen; Qualitätsbewusstsein, Akribie.</p>		
<p>Arbeitsaufwand: 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>Voraussetzungen: Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia Grundlagen II (INF-0166) - empfohlen</p>		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Multimedia I: Usability Engineering (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4</p>
<p>Inhalte: Methoden, Werkzeuge und Vorgehensweisen zur Gestaltung von gebrauchstauglichen Softwareprodukten</p>
<p>Literatur:</p> <ul style="list-style-type: none"> • Ben Shneiderman, "Designing the User Interface: Strategies for Effective Human-Computer Interaction" • Jakob Nielsen, "Usability Engineering" • Helen Sharp, Yvonne Rogers und Jenny Preece, "Interaction Design beyond Human Computer Interaction"
<p>Zugeordnete Lehrveranstaltungen: Multimedia 1: Usability Engineering (Vorlesung) Durch die zunehmende Verbreitung von Computern in allen Lebensbereichen und deren Einbettung in die natürliche Umgebung des Benutzers wird die Gestaltung der Mensch-Technik-Interaktion zu einer großen Herausforderung. Während bislang rein technische Aspekte dominierten, geht der Trend zu Entwurfsprozessen, die den Nutzer in den Vordergrund stellen und ihn bereits in der Entwurfsphase miteinbeziehen. Ziel dieser</p>

Veranstaltung ist die praxisnahe Vermittlung von Wissen zum Thema Usability Engineering. Dabei werden unter anderem unterschiedliche Verfahren zur nutzerzentrierten Entwicklung von klassischen und neuartigen Nutzerschnittstellen vorgestellt. Der praktische Teil der Vorlesung startet mit der Entwicklung eines Konzeptes und endet mit der Implementierung eines ersten klickbaren Prototypen.

Modulteil: Multimedia I: Usability Engineering (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Multimedia 1: Usability Engineering (Übung)

Prüfung

schriftliche Abgaben

Übung + Praktikum

Modul INF-0178: Praktikum Usability Engineering		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Die Studierenden sind mit Methoden und Techniken des Usability Engineering vertraut. Nach der erfolgreichen Teilnahme verfügen sie über die notwendigen Kenntnisse, um nach den Vorgaben des Nutzerzentrierten Designprozesses Anwendungsszenarien zu analysieren und zielgruppengerechte Softwarelösungen zu entwerfen. Sie sind dazu fähig, aktuelle Interaktionsparadigmen und Design-Richtlinien in Modelle und Programme für neuartige Interaktionsgeräte zu übersetzen, sowie sich selbstständig in die notwendigen Technologien einzuarbeiten. Des weiteren können sie praxisrelevante Evaluationsmethoden anwenden, um die Qualität des erstellten Softwareprototypen zu bewerten. 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>Schlüsselqualifikationen: 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>Arbeitsaufwand: Gesamt: 240 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 90 Std. Praktikum (Präsenzstudium)</p>		
Voraussetzungen: Modul Multimedia I: Usability Engineering (INF-0175) - empfohlen		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Praktikum Usability Engineering Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		
Inhalte: Die konkrete Aufgabenstellung für Studentenprojekte wird jedes Jahr neu entworfen.		
Literatur: Literaturhinweise werden je nach Thema zu Beginn des Semesters bekanntgegeben.		
<p>Prüfung Vortrag mit Softwarepräsentation, Ausarbeitung mit Softwaredokumentation Projektarbeit</p>		

Modul INF-0179: Einführung in die Spieleprogrammierung		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul verstehen die Studierenden die wesentlichen Konzepte zur Entwicklung eines Spiels. Sie sind in der Lage, eigene Gameplay-Konzepte umzusetzen und passende KI-Verfahren zu Entscheidungsfindung sowie Ablaufsteuerung einzusetzen. Sie verfügen über die Kenntnisse der Denkweise und die Sprache anwendungsrelevanter Disziplinen. Im Rahmen der Vorlesung erlernen sie, KI-Komponenten mit geeigneten Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmik selbstständig zu entwickeln und technisch umzusetzen. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen sowie zum logischen, analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikationen: Auswahl und sichere Anwendung geeigneter Methoden, Fachübergreifende Kenntnisse, Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen, Fertigkeit zur Entwicklung und Umsetzung von Lösungsstrategien für komplexe Probleme, Verstehen von Teamprozessen, Fertigkeit der Zusammenarbeit in Teams, Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen, Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 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) 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)</p>		
<p>Voraussetzungen: Ferienaufgabe</p>		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Einführung in die Spieleprogrammierung (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: Game Engines, Entscheidungsfindung für KI-Charaktere, Wegfindung und Navigation, Gruppenverhalten und Gruppendynamik, Animationen, Physik, Storytelling, Ein-/Ausgabemethodik, Audio, Game Design, Grafik und Shaderprogrammierung</p>		
<p>Literatur: Skript</p>		
<p>Modulteil: Einführung in die Spieleprogrammierung (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 4</p>		

Prüfung

Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation, Übungsaufgaben

Projektarbeit

Modul INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul sind die Studierenden mit den Grundkonzepten der Signalverarbeitung und des maschinellen Lernens vertraut. Sie sind in der Lage, fachliche Lösungskonzepte in Programme und Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Im Rahmen des Praktikums erlernen sie außerdem, 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. Besonders gefördert wird zudem die Fähigkeit geeignete Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmik selbstständig zu entwickeln und technisch umzusetzen.</p> <p>Schlüsselqualifikationen: Fortgeschrittene Kenntnisse der Signalverarbeitung und des maschinellen Lernens, Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen, Fertigkeit zur Entwicklung und Umsetzung von Lösungsstrategien für komplexe Probleme, Verstehen von Teamprozessen, Fertigkeit der Zusammenarbeit in Teams, Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen, Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p>Voraussetzungen: Programmiererfahrung Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia Grundlagen II (INF-0166) - empfohlen</p>		
<p>Angebotshäufigkeit: jedes Sommersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Moduleile</p>		
<p>Moduleil: Praktikum Multimodale Echtzeitsignalverarbeitung Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		
<p>Inhalte: Die konkrete Aufgabenstellung aus dem weiten Gebiet der multimodalen Echtzeitsignalverarbeitung wird jedes Jahr neu entworfen.</p>		
<p>Literatur: Literaturhinweise werden je nach Thema zu Beginn des Semesters bekanntgegeben.</p>		
<p>Prüfung Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation Projektarbeit</p>		

Modul INF-0183: Praktikum Spieleprogrammierung		8 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul verstehen die Studierenden die wesentlichen Konzepte der Spieleentwicklung. 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 zu präsentieren. Sie sind in der Lage, fachliche Lösungskonzepte in Programme und Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Sie verfügen über die Kenntnisse der Denkweise und die Sprache anwendungsrelevanter Disziplinen. Im Rahmen des Praktikums erlernen sie, verschiedenste Komponenten eines Spiels mit geeigneten Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmik selbstständig zu entwickeln und technisch umzusetzen. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen sowie zum logischen, analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikationen: 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 bedeutendetechnischen Entwicklungen; Qualitätsbewusstsein, Akribie.</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
Voraussetzungen: Modul Einführung in die Spieleprogrammierung (INF-0179) - empfohlen		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 2.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Praktikum Spieleprogrammierung Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		
<p>Inhalte: Innerhalb des Praktikums soll ein Spiel entwickelt werden. Der inhaltliche Schwerpunkt des Praktikums wird jedes Jahr neu festgelegt.</p>		
<p>Literatur: Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.</p>		
<p>Zugeordnete Lehrveranstaltungen: Praktikum Spieleprogrammierung (Praktikum) Ziel der Veranstaltung ist das Wissen aus der Vorlesung "Einführung in die Spieleprogrammierung" praktisch anzuwenden und zu vertiefen. Voraussetzung ist deshalb die erfolgreiche Teilnahme an dieser Veranstaltung! Im Verlauf des Praktikums werden die Teilnehmer gemeinsam ein Spiel entwickeln. Das genaue Thema des Spiels wird später festgelegt. Die Teilnehmer des Praktikums werden in kleinere Gruppen aufgeteilt, die an</p>		

unterschiedlichen Aufgabenbereichen arbeiten, wie der Visualisierung des Spielgeschehens, der Interaktion des Nutzers mit der Anwendung oder der künstlichen Intelligenz der virtuellen Charaktere. Um ein funktionierendes Spiel zu entwickeln, ist neben der Arbeit in der jeweiligen Gruppe auch zwingend gruppenübergreifende Zusammenarbeit gefordert, z. B. zur Absprache der Schnittstellen und Integration der einzelnen Komponenten. Die Teilnehmerzahl ist aufgrund der Raumgröße und Hardwareverfügbarkeit auf max. 20 Studenten beschränkt. ... (weiter siehe Digicampus)

Prüfung

Vortrag mit Softwarepräsentation, Ausarbeitung mit Softwaredokumentation

Projektarbeit

Modul INF-0189: Qualitätssicherung im Software Engineering		5 ECTS/LP
Version 2.0.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen: Nach erfolgreicher Absolvierung der Lehrveranstaltung verstehen Studierende die Grundtechniken der Qualitätssicherung und der Problemstellungen des Testens in der Praxis. Sie sind in der Lage, SW-Module zu spezifizieren und kennen die wesentlichen Testtechniken und deren Einsatzzwecke im Software Engineering. Sie können Testkonzepte für große Softwaresysteme entwerfen, modellieren und anwenden. Die Studierenden sind für das Thema Qualität im Software Engineering sensibilisiert und können verschiedene Qualitätskriterien/-metriken kritisch hinterfragen und bewerten.</p> <p>Des Weiteren kennen und verstehen sie die Prinzipien von konstruktiven Qualitätssicherungstechniken und -praktiken.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Training des logischen Denkens • Analytisch-methodische Kompetenz • Abwägen von Vorgehensweisen • Erwerb von Abstraktionsfähigkeiten • Verbesserung der eigenen Softwareentwicklungskompetenz 		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Qualitätssicherung im Software Engineering (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		

Inhalte:

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.

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 (Mündliche Prüfung)**

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0198: Intelligente Systeme		5 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
Lernziele/Kompetenzen: 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.		
Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken		
Arbeitsaufwand: Gesamt: 150 Std. 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) 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)		
Voraussetzungen: Sinnvoll sind Kenntnisse und Interesse an: Diskreten Strukturen und Logiken		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Intelligente Systeme (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: <ul style="list-style-type: none"> • Wissensrepräsentation, Sprachen, Ontologien (Prädikatenlogik, Hornlogik, Semantische Netze, ...) • Der Inferenzbegriff ("Beweis"kalküle, Resolution) • Klassifikation und Konzepte (Clustering, Hierarchical Clustering, Incremental Clustering) • Konzeptanalyse und Konzepthierarchien (Formal Concept Analysis) • Relationale und Soft-Computing Ansätze zur Datenanalyse (Grobmengen, Fuzzy-Logik, ...) • Generalisierung und Induktion (Subsumption, Negation, Abduktion, ...) • Lernen: Finden von Hypothesen (Induktives Logisches Programmieren, Bagging) 		
Literatur: <ul style="list-style-type: none"> • Eigenes Skriptum • S. Russell & P. Norvig: Artificial Intelligence - A Modern Approach, 3rd Edition, 2010 • weitere Literatur wird in der Vorlesung angegeben 		
Modulteil: Intelligente Systeme (Übung)		
Lehrformen: Übung		
Sprache: Deutsch		
SWS: 2		
Prüfung		
Intelligente Systeme (Klausur) Klausur / Prüfungsdauer: 90 Minuten		

Modul INF-0201: Platzeffiziente Algorithmen		5 ECTS/LP
Version 1.0.0 (seit WS15/16) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<p>Lernziele/Kompetenzen: 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>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Fähigkeit zur präzisen fachlichen Ausdrucksweise; Beherrschung effektiver Lern- und Arbeitstechniken; Abstraktionsfähigkeit; Problemlösungskompetenz; Qualitätsbewusstsein.</p>		
<p>Arbeitsaufwand: 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>		
<p>Voraussetzungen: Teilnehmer der Vorlesung sollten über gute algorithmische Kenntnisse verfügen, insbesondere im Bereich Graphenalgorithmen. Modul Informatik 3 (INF-0111) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 4</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Platzeffiziente Algorithmen (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: 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>Literatur:</p> <ul style="list-style-type: none"> • Skript 		

Modulteil: Platzeffiziente Algorithmen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Platzeffiziente Algorithmen (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Modul INF-0207: Reinforcement Learning		8 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul verstehen die Studierenden die wesentlichen Konzepte des Reinforcement Learnings. Sie sind in der Lage, fachliche Lösungskonzepte in Programme und Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Sie verfügen über die Kenntnisse der Denkweise und die Sprache anwendungsrelevanter Disziplinen. Im Rahmen der Vorlesung erlernen sie, lernende Komponenten mit geeigneten Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmen selbstständig zu entwickeln und technisch umzusetzen. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen sowie zum logischen, analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikationen: Fortgeschrittene mathematisch-formale Methodik, Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen, Fertigkeit zur Entwicklung und Umsetzung von Lösungsstrategien für komplexe Probleme, Verstehen von Teamprozessen, Fertigkeit der Zusammenarbeit in Teams, Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen, Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 60 Std. Übung (Präsenzstudium) 120 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)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Reinforcement Learning (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: Markov-Entscheidungsproblem, Dynamische Programmierung, Monte Carlo Methoden, TD-Lernen, Eligibility Traces, Hierarchisches Bestärkendes Lernen, Planen und Lernen, Generalisierung		
Literatur: <ul style="list-style-type: none"> • Skript • Richard S. Sutton und Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998 		
Zugeordnete Lehrveranstaltungen:		
Reinforcement Learning (Vorlesung) Inhalt der Veranstaltung sind Methoden und Prinzipien des Reinforcement Learning. In der Übung werden verschiedene Probleme modelliert, mit Hilfe von Lernalgorithmen als Simulationen implementiert und verglichen. Siehe https://www.informatik.uni-augsburg.de/lehrstuehle/hcm/lectures/2017ws/rl/		

Modulteil: Reinforcement Learning (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Reinforcement Learning (Übung)

Prüfung

Reinforcement Learning (Projektarbeit)

Projektarbeit, Projektarbeit / mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0213: Parallele und Verteilte Datenbanksysteme		5 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: PD Dr. Markus Endres		
<p>Lernziele/Kompetenzen: 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 im Detail zu verstehen, zu bewerten und anzuwenden. Darüber hinaus können die Studierenden weiterführende komplexe Problemstellungen auf dem Gebiet Datenbanken, insbesondere unter Verwendung von parallelen und verteilten Algorithmen, analysieren und lösen. Die Vorlesung vermittelt zudem die Fertigkeit zum logischen, analytischen und konzeptionellen Denken. Bei den in der Veranstaltung anfallenden praxisnahen Programmieraufgaben ist eine sichere und überzeugende Darstellung von Ideen und Konzepten sowie die Zusammenarbeit in Teams entscheidend.</p> <p>Schlüsselqualifikationen: Fachspezifische Vertiefung; Kenntnisse der Vor-/Nachteile von Entwurfalternativen und deren Bewertung im jeweiligen Anwendungszusammenhang; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen; Fähigkeit, vorhandenes Wissen selbständig zu erweitern;</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 3.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Parallele und Verteilte Datenbanksysteme (Vorlesung)		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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>Literatur:</p> <ul style="list-style-type: none"> • - Taniar et al.: High-Performance Parallel Database Processing and Grid Databases • - E. Rahm: Mehrrechner-Datenbanksysteme • - M. T. Özsu, P. Valduriez: Principles of Distributed Database Systems • - P. Dada: Verteilte Datenbanken und Client/Server-Systeme • - S. Conrad: Förderierte Datenbanksysteme • - S. K. Rahimi, F. S. Haug: Distributed Database Management Systems - A Practical Approach 		

Modulteil: Parallele und Verteilte Datenbanksysteme (Übung)

Sprache: Deutsch

SWS: 2

Prüfung

Parallele und Verteilte Datenbanksysteme (mdl. Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0216: Vertiefte Multicore-Programmierung		8 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen auf einem fortgeschrittenen, wissenschaftlichen Niveau in folgenden Bereichen: Parallelprogrammierung von speichergekoppelten Systemen in C++11 und OpenMP, Entwurf von lock-freien Algorithmen, Programmierung eines Hardware-Transaktionsspeicher-Systems am Beispiel von Intel TSX, nachrichtengekoppelte Programmierung mit dem Message-Passing-Interface (MPI), Programmierung von GPU-Beschleunigerkarten mit Nvidia CUDA.</p> <p>Sie können die Vor- und Nachteile der unterschiedlichen Programmiermodelle und Architekturen einschätzen, und parallele Programme in den jeweiligen Sprachen analysieren. Durch die praktischen Übungen setzen die die Studierenden die erlernten Konzepte direkt in die Praxis um. Dabei erwerben die Studierenden fortgeschrittene Programmierkenntnisse in C++11, Intel TSX, MPI, OpenMP und CUDA. Sie können selbständig das Laufzeitverhalten der jeweiligen Programmiermodelle bewerten und optimieren. In einer abschließenden Projektphase lernen die Studierenden, eine komplexe Aufgabenstellung im Team zu planen, nach einem selbst entwickelten Lösungsansatz zu bearbeiten und die Resultate in wissenschaftlicher Weise im Plenum zu diskutieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Analytische-methodische Kompetenz im Bereich der Multicore-Programmierung von CPUs und GPUs; Projektgebundene Arbeit und Zeitmanagement; Auswahl und sichere Anwendung geeigneter Methoden; Fähigkeit, vorhandenes Wissen selbständig zu kombinieren und zu erweitern; Selbstreflexion</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 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) 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Moduleile		
Modulteil: Vertiefte Multicore-Programmierung (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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.</p>		

Literatur:

- M. Herlihy, N. Shavit: The Art of Multiprocessor-Programming, Morgan Kaufmann, 2012, ISBN: 978-0123973375
- M. McCool, J. Reinders, A. D. Robison: Structured Parallel Programming: Patterns for Efficient Computation, Morgan Kaufmann, 2012, ISBN: 978-0124159938
- T. Rauber, G. Runger: Parallele Programmierung, Springer Verlag, 2007, ISBN: 978-3540465492
- es werden die jeweils neuesten APIs/Unterlagen aus dem Internet verwendet

Modulteil: Vertiefte Multicore-Programmierung (ubung)

Lehrformen: ubung

Sprache: Deutsch

SWS: 4

Inhalte:

Ziel der ubung ist es, den Umgang mit den unterschiedlichen Programmiermodellen sowie Performanzanalyse- und Debugging-Techniken in praktischen Beispielen zu vertiefen. Die ubung wird durch eine Projektphase abgeschlossen, die es den Studierenden ermoglicht, die behandelten Programmier Techniken in einem umfangreicheren Projekt selbstandig anzuwenden, Ergebnisse auszuwerten und zu prasentieren.

Prufung

Vertiefte Multicore-Programmierung (mundliche Prufung)

Mundliche Prufung, Dauer: 30-45 Minuten

Beschreibung:

Projektvorstellung und -abnahme, Fragen zu Vorlesung und ubung

Modul INF-0217: Praktikum Autonomes Fahren		10 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Praktikum Autonomes Fahren verstehen es die Studierenden praxisnahe Problemstellungen hoher Komplexität im Bereich der Konzeptionierung, Entwicklung und Absicherung von hochautomatisierten/ autonomen Fahrzeugen mit aktuellen Methoden und Tools der modellbasierten Entwicklung zu lösen. Die Studierenden erlangen tiefgehende fachspezifische als auch fächerübergreifende Kenntnisse und Fähigkeiten, beispielsweise aus der Hardwarenahen Informatik, dem Software Engineering, als auch der zugrundeliegenden Fahrphysik und Mathematik. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren, zu präsentieren und verständlich zu dokumentieren.</p> <p>Schlüsselqualifikation: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; Verständliche, sichere und überzeugende Präsentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein; Kommunikationsfähigkeit; Fertigkeit der Zusammenarbeit in Teams und Verstehen von Teamprozessen; Projektmanagementfähigkeiten</p>		
<p>Arbeitsaufwand: Gesamt: 300 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 150 Std. Praktikum (Präsenzstudium)</p>		
<p>Voraussetzungen: 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>		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 10	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Praktikum Autonomes Fahren Lehrformen: Praktikum Sprache: Deutsch / Englisch SWS: 10</p>		

Inhalte:

In diesem Praktikum lernen die Teilnehmer, wie verschiedene ausgewählte Teilaspekte des Autonomen Fahrens umgesetzt, simuliert und analysiert werden können.

In einem Einführungskurs werden die benötigten theoretischen Grundlagen für die Bearbeitung der Praxisaufgaben gelegt.

Darüber hinaus lernen die Teilnehmer u.a. das im Automotive-Umfeld häufig eingesetzte Entwicklungswerkzeug „ADTF“ von Elektrobit und die darin verwendete Programmiersprache C++ kennen.

Nach dem Einführungskurs sollen in Gruppen von zwei bis drei Teilnehmern mithilfe der genannten Werkzeuge autonome Fahrfunktionen umgesetzt werden.

Die entwickelten Ergebnisse werden abschließend an 1:8 Fahrzeugmodellen, deren Sensorik realen Fahrzeugen sehr nahe kommt, demonstriert und ausgewertet.

Prüfung

Praktikum Autonomes Fahren

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Modul INF-0233: Industrierobotik		8 ECTS/LP
Version 2.1.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen: Die Studierenden sind in der Lage Industrieroboter zu programmieren, kennen das dafür erforderliche methodische Vorgehen und können die Eignung verschiedener Vorgehensweise bewerten. 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.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> Analytisch-methodische Kompetenz Abwägen von Lösungsansätzen Erwerb von Abstraktionsfähigkeiten 		
<p>Bemerkung: Dieses Modul ersetzt das Modul "INF-0132: Software in Mechatronik und Robotik". Sofern dieses bereits belegt wurde, ist eine erneute Belegung nicht möglich.</p>		
<p>Arbeitsaufwand: 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>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Industrierobotik (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: 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.</p>		
<p>Literatur:</p> <ul style="list-style-type: none"> L. Sciavicco, B. Siciliano: Modelling and Control of Robot Manipulators. Reihe: Advanced Textbooks in Control and Signal Processing. Springer 2000 (2nd Ed.) Handbücher von KUKA Folienhandout 		
Zugeordnete Lehrveranstaltungen:		

Industrierobotik (Vorlesung)

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

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 3 Monate

Modul INF-0235: Software für Industrie 4.0		5 ECTS/LP
Version 1.1.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
Lernziele/Kompetenzen: Die Studierenden sind in der Lage die praktischen und methodischen 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 und kennen so unterschiedliche anwendungsrelevante Disziplinen. Sie sind in der Lage Lösungskonzepte zu erstellen und dabei Entwurfsalternativen bewerten und anwenden.		
Schlüsselqualifikationen: <ul style="list-style-type: none"> • Analytisch-methodische Kompetenzen • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten 		
Arbeitsaufwand: Gesamt: 150 Std. 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) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Software für Industrie 4.0 (Vorlesung)		
Lehrformen: Vorlesung		
Sprache: Deutsch		
SWS: 2		
Inhalte: Die Vorlesung behandelt aktuelle Themen im Kontext von Industrie 4.0. Dazu zählen folgende Inhalte: <ul style="list-style-type: none"> • Einführung in die Automatisierung • Referenzarchitekturen für Industrie 4.0 • Einführung in die (mobile Service) Robotik • OPC UA • AutomationML • Data Analytics für Industrie 4.0 		
Literatur: <ul style="list-style-type: none"> • 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

Software für Industrie 4.0

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Modul INF-0243: Process Mining <i>Process Mining</i>		6 ECTS/LP
Version 1.1.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Robert Lorenz		
<p>Lernziele/Kompetenzen: Teilnehmer verstehen die folgenden wesentlichen Konzepte der Informatik auf einem vertieften wissenschaftlichen Niveau mit ihren mathematisch-formalen Grundlagen: Ereignis-Log, Halbordnung und partielle Sprache, Petrinetz, Nebenläufigkeit, sequentielle und kausale Semantik eines nebenläufigen Systems, Synthese von nebenläufigen Systemen, Geschäftsprozess, Process Mining, Process Discovery. Sie können nebenläufige Systeme mittlerer Komplexität in einer geeigneten Modellierungssprache modellieren, sowie zu einem gegebenen nebenläufigen System verschiedene Verhaltensmodelle zur Analyse und Simulation generieren, bewerten und ineinander überführen. Sie kennen die verschiedenen Process Mining Anwendungsfälle mit dazu passenden Lösungstechniken und Qualitätskriterien zu deren Bewertung. Teilnehmer verfügen über eine fortgeschrittene mathematisch formale Methodik für die Analyse und Formalisierung komplexer Process Discovery Probleme. Sie sind in der Lage, geeignete Algorithmen zu deren Lösung bzgl. qualitativer und quantitativer Kriterien sicher begründet auszuwählen, sowie diese anzuwenden und die Ergebnisse zu interpretieren und zu bewerten.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen, konzeptionellen und algorithmischen Denken; Eigenständiges Arbeiten mit Lehrbüchern und englischsprachiger Fachliteratur; Verständliche Präsentation von Ergebnissen; Qualitätsbewußtsein; Akribie; Grundsätze guter wissenschaftlicher Praxis; Wissenschaftliche Methodik;</p>		
<p>Arbeitsaufwand: Gesamt: 180 Std. 45 Std. Vorlesung (Präsenzstudium) 15 Std. Übung (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>		
Voraussetzungen: keine		ECTS/LP-Bedingungen: Bestehen der Modulprüfung
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Process Mining (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch / Englisch SWS: 3</p>
<p>Inhalte: Die Veranstaltung gibt einen fundierten Überblick über das Forschungsgebiet Process Mining:</p> <ul style="list-style-type: none"> - Modelling Techniques: Workflow Nets, Partial Languages - Discovering algorithms - Conformance Checking - Process Enhancement - The PROM Framework

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

Modulteil: Process Mining (Übung)

Lehrformen: Übung

Sprache: Deutsch / Englisch

SWS: 1

Prüfung

Process Mining (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0248: Kollaborative Robotik		8 ECTS/LP
Version 1.1.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen: Die Studierenden sind in der Lage kollaborative Roboter zu programmieren und verstehen die regulatorischen Randbedingungen (vgl. ISO/TS 15066:2016). Sie verstehen 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. Die Studierenden können abwägen, für welche praxisrelevanten Problemstellungen kollaborative Roboter eingesetzt werden können.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten • Erwerb von Grundlagen für die Sicherheitsbewertung 		
<p>Bemerkung: Voraussetzungen (empfohlen): Software für Industrie 4.0 oder Industrierobotik oder Grundlagen der Robotik</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std.</p>		
<p>Voraussetzungen: keine</p>		<p>ECTS/LP-Bedingungen: Bestehen der Modulprüfung</p>
<p>Angebotshäufigkeit: unregelmäßig</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Kollaborative Robotik (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 4</p>		

Inhalte:

Die Digitalisierung dringt mittlerweile in viele Bereiche des täglichen Lebens vor. Von Industriemaschinen und Robotersystemen (*Smart Factory, Industrie 4.0*) bis hin zu intelligenten Hausgeräten oder Heizungsanlagen (*Smart Home*) 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.

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.

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.

Infolgedessen liegt der thematische Fokus der Veranstaltung auf

- der Umsetzung multi-modaler Benutzerschnittstellen,
- der Gestaltung neuartiger, auch physischer Interaktionsmöglichkeiten,
- der Realisierung intelligenter Verhaltensweisen durch Techniken der Künstlichen Intelligenz bzw. des Machine Learnings und
- der Entwicklung kognitiver Fähigkeiten des Robotersystems.

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.

Zugeordnete Lehrveranstaltungen:**Übung zu Kollaborative Robotik (Übung)****Modulteil: Kollaborative Robotik (Vorlesung)**

Lehrformen: Vorlesung

Sprache: Deutsch

SWS: 2

Literatur:

- ISO/TS 15066:2016. 2016-02-15. Robots and robotic devices - Collaborative robots
- 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
- DIN EN ISO 10218-1:2012. 01.2012. Industrieroboter - Sicherheitsanforderungen - Teil 1: Roboter (ISO 10218-1:2011); Deutsche Fassung EN ISO 10218-1:2011

Zugeordnete Lehrveranstaltungen:**Kollaborative Robotik (Vorlesung)**

Der Schwerpunkt der Veranstaltung liegt auf der Entwicklung von Software für kollaborative (mobile) Manipulatoren; die grundlegende Hardware, d.h. die Roboter, mobilen Plattformen, Greifer und Sensoren, wird

bereitgestellt. 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. Infolgedessen liegt der thematische Fokus der Veranstaltung auf • der Umsetzung multi-modaler Benutzerschnittstellen, • der Gestaltung neuartiger, auch physischer Interaktionsmöglichkeiten, • der Realisierung intelligenter Verhaltensweisen durch Techniken der Künstlichen Intelligenz bzw. des Machine Learnings und • der Entwicklung kognitiver Fähigkeiten des Robotersystems. Diese Kernelemente bieten die Grundlage, um eine praktische Aufgabenstellung prototypisch umzusetzen. Die Programmierung erfolgt dab
... (weiter siehe Digicampus)

Prüfung

Kollaborative Robotik

praktische Prüfung / Prüfungsdauer: 60 Minuten

Bearbeitungsfrist: 3 Monate

Modul INF-0250: Praktikum Reinforcement Learning		8 ECTS/LP
Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul verstehen die Studierenden die wesentlichen Konzepte des Reinforcement Learnings. 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 zu präsentieren. Sie sind in der Lage, fachliche Lösungskonzepte in Programme und Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Sie verfügen über die Kenntnisse der Denkweise und die Sprache anwendungsrelevanter Disziplinen. Im Rahmen des Praktikums erlernen sie, lernende Komponenten mit geeigneten Methoden wissenschaftlich aussagekräftig zu bewerten, die Methoden und Algorithmen selbstständig zu entwickeln und technisch umzusetzen. Besonders gefördert wird in diesem Rahmen auch die Fertigkeit zur sicheren und überzeugenden Darstellung von Ideen und Konzepten, verständlichen Darstellung und Dokumentation von Ergebnissen sowie zum logischen, analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikationen: 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>Arbeitsaufwand: Gesamt: 240 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 90 Std. Praktikum (Präsenzstudium)</p>		
<p>Voraussetzungen: Programmiererfahrung Modul Reinforcement Learning (INF-0207) - empfohlen</p>		
<p>Angebotshäufigkeit: jedes Sommersemester</p>	<p>Empfohlenes Fachsemester: ab dem 2.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Praktikum Reinforcement Learning Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>		
<p>Inhalte: Die konkrete Aufgabenstellung aus dem Gebiet "Reinforcement Learning" wird jedes Semester neu entworfen.</p>		
<p>Prüfung Vortrag mit Softwarepräsentation, Ausarbeitung mit Softwaredokumentation Projektarbeit</p>		

Modul INF-0272: Intelligente Signalanalyse in der Medizin		5 ECTS/LP
Version 1.1.0 (seit WS17/18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Knowledge: The students learn the principal concepts of sequential signal processing, signal source separation, and feature extraction and information reduction exemplified by medically relevant audio and bio signals. They further gain insight into machine learning principles such as learning dynamics and context as is needed for many intelligent signal analysis tasks. They will learn about different problems and solutions in the analysis of a variety of signals relevant in the context of health care, wellbeing, and general medical signals analysis. Students will get to know the mindset of modern machine learning, computer-aided health care, and get to know ethical implications.</p> <p>Skills: The students will be able to choose appropriate algorithms of signal processing and machine intelligence, further develop these, design new solutions, and apply these to a broad range of medical signal analysis problems. They will practice to think logically and conceptionally in order to select appropriate solutions to a given task. Students will be able to recognise important technical developments in the field of signal processing, machine learning and e-Health/m-Health.</p> <p>Competences: The students are able to characterise, judge on the quality and suitability, and design suited algorithmic solutions for intelligent signal analysis with a focus on medical signals. They are further able to realise the learnt concepts in programs and machine learning models. Participants will be able to analyse and structure complex and practice-oriented problems in the field of m-Health and e-Health and to find suitable and state-of-the-art solutions. They know how to make scientifically meaningful evaluations of proposed systems. They will further learn how to document and present results in a reasonable and meaningful way.</p> <p>Key skills: Formal methods; Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality awareness; Scientific working; Literature research.</p>		
<p>Arbeitsaufwand: 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>		
<p>Voraussetzungen: Knowledge of basic mathematic lectures should be present.</p>		
<p>Angebotshäufigkeit: jedes Sommersemester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 4</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Intelligente Signalanalyse in der Medizin (Vorlesung) Lehrformen: Vorlesung Sprache: Englisch Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe) SWS: 2</p>		

Inhalte:

Topics: Basics of Signal Processing, Signal Source Separation, Data Acquisition and Annotation, Audio-Visual Feature Extraction, Machine Learning, e-Health, m-Health, Ethics, Python, Machine Learning Toolkits.

Literatur:

Björn Schuller: „*Intelligent Audio Analysis*“, Signals and Communication Technology, Springer, ISBN: 978-3642368059, 2013.

Modulteil: Intelligente Signalanalyse in der Medizin (Übung)

Lehrformen: Übung

Sprache: Englisch

Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)

SWS: 2

Prüfung

Intelligente Signalanalyse in der Medizin (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0273: Praktikum Mobile Sensing for Fitness and Well-being		5 ECTS/LP
Version 1.0.0 (seit WS17/18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen: Students learn to realise concepts and models of sensing health- and fitness-related parameters on smart devices. They learn how to acquire signals from different modalities and sensors and to implement algorithms of pattern recognition on mobile devices. After participation in the Praktikum, students know how to analyse and structure complex problems in the field and to select suitable and state-of-the-art approaches to their solution.</p> <p>Participants are trained in their analytical and conceptional skills as well as in practical programming skills to transfer their knowledge to a practical task. They learn how to make scientifically meaningful assessments of their system using appropriate methods. They will know the mindset of two different fields, software development and machine learning. All knowledge obtained during the Praktikum is applied in practice-oriented tasks. Furthermore, after participation, students will be able to recognise important technical evolution in the field of sensors, mobile apps and related tools, and intelligent signal analysis.</p> <p>Students will work in teams and organise their work and task distribution in an autonomous way. They will learn how to summarise, present and document results in a reasonable way</p> <p>Key skills: Formal methods; Methods for software development and abstraction; Versioning tools; Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams, Understanding of team management; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality Awareness.</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Praktikum Mobile Sensing for Fitness and Wellbeing		
Lehrformen: Praktikum		
Sprache: Englisch		
SWS: 4		
<p>Inhalte: Research in the field of m-Health is focussed on the design and development of sensors, systems, and applications to recognise, interpret and simulate human states w.r.t. fitness, health, and wellbeing. In this Praktikum, students will experience in designing relevant systems, which are using modalities originating from different sensors, such as, vital signs, audio, speech, and video. In small teams, they will implement and evaluate an application running on a smart device.</p>		
<p>Literatur: Wird vom Dozenten / von der Dozentin bekannt gegeben</p>		

Zugeordnete Lehrveranstaltungen:

Praktikum Mobile Sensing for Fitness and Wellbeing (Praktikum)

This course will be given as a block course. The schedule and the location are going to be announced later.
Please NOTE that you need to register via STUDIS in order to participate in the Praktikum.

Prüfung

Praktikum Mobile Sensing for Fitness and Wellbeing

Praktikum

Modul INF-0277: Analyzing Massive Data Sets		8 ECTS/LP
Version 1.1.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Peter Michael Fischer		
<p>Lernziele/Kompetenzen: Nach der Teilnahme an der Veranstaltung sind die Studierenden in der Lage die Konzepte und Methoden, Verfahren, Techniken und Technologien zur Analyse von massiv großen Datensätzen zu verstehen und zu bewerten. Mögliche Inhalte umfassen:</p> <ul style="list-style-type: none"> • Grundlagen des Information Retrieval • Ähnlichkeitssuche und Clustering • Analyse von Datenströmen und temporalen Daten • Webgraphen: Linkanalyse und soziale Netzwerke • Dynamische Netzwerke und Informationsausbreitung • Empfehlungssysteme und Onlinewerbung • Berechnungsverfahren für massive Datensätze <p>Außerdem können die Studierenden fachliche Lösungskonzepte zur Analyse großer Datensätze in Programme umsetzen.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten; Fachspezifische Vertiefung; Umsetzen fachlicher Lösungskonzepte in Programme und Modelle; Kenntnisse der Vor-/Nachteile von Entwurfsalternativen und Bewertung im jeweiligen Anwendungszusammenhang; Auswahl und sichere Anwendung geeigneter Methoden; Fertigkeit, mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen anzustellen; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen;</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium)</p>		
<p>Voraussetzungen: Modul Datenbanksysteme (INF-0073) - empfohlen Modul Diskrete Strukturen für Informatiker (INF-0109) - empfohlen Modul Informatik 3 (INF-0111) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Analyzing Massive Data Sets (Vorlesung) Lehrformen: Vorlesung Sprache: Englisch SWS: 4</p>		

Inhalte:

Die Vorlesung behandelt grundlegende Konzepte zur Analyse massiv großer Datensätze wie Informationsextraktion, Ähnlichkeitssuche, Clustering, Link- und Netzwerkanalyse sowie deren Implementierung.

Literatur:

- Mining of Massive Datasets. J. Leskovec, A. Rajaraman, J.D. Ullman. Cambridge University Press, 2014
- D. Easley, J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, 2010.
- R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval

Weitere Literatur wird in der Vorlesung bekannt gegeben

Modulteil: Analyzing Massive Data Sets (Übung)

Lehrformen: Übung

Sprache: Englisch / Deutsch

SWS: 2

Prüfung

Analyzing Massive Data Sets

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0279: Music Informatics		5 ECTS/LP
Version 1.2.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Knowledge: The course Music Informatics presents the fundamental concepts of music theory and the music language and its representation in the visual, symbolic, and acoustic domain. Several digital formats for music symbolic representation, such as Music XML, MEI, Kern**, and MIDI protocol, as well as open source tools such as LilyPond and Csound will be introduced. Machine learning principles and techniques with applications in music information retrieval and computational musicology will be practically applied. Students will learn about different problems and solutions in the analysis of symbolic and acoustic music data. Students will get to know the mindset from both sides, the musicological and the computer scientist perspective.</p> <p>Skills: The students will understand the basic principles of music theory and its representation in digital language, being able to analyse, interpret, and create musical samples in a variety of symbolic formats and programming languages. They will learn to apply machine learning procedures, such as feature extraction and pattern recognition, to music information retrieval problems, such as key detection and music-score synchronisation, amongst other. After participation, students will know how to advance existing concepts and approaches in the field of music informatics and data analysis. Furthermore, they will be able to recognise important technical developments in the field of data science and signal processing.</p> <p>Competences: By integrating basic principles of music theory, its representation in digital language, and machine learning techniques, the students will be able to identify new problems and solutions in the field of music information retrieval considering a variety of musical styles and genres. The students are able to characterise, judge on the quality and suitability, and design suited algorithmic solutions for music data analysis in both the symbolic and the audio domain.</p> <p>Participants will be able to analyse and structure complex and practice-oriented problems in the field of music informatics and to find suitable solutions, by using state-of-the-art tools and complementary methods, if needed. They know how to make scientifically meaningful evaluations of proposed systems. They will further learn how to document and present results in a reasonable and meaningful way.</p> <p>Key skills: Computational musicology, Music theory, Digital Music Representation, Basics of Signal Processing, Machine Learning, Music Information Retrieval, Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality awareness; Scientific working; Literature research.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std.</p> <p>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> <p>60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p>		
Voraussetzungen:		
Knowledge of basic mathematic lectures should be present		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	

Moduleile
Moduleil: Music Informatics (Lecture) Lehrformen: Vorlesung Sprache: Englisch SWS: 2
Inhalte: In Music Informatics, the basic principles of music theory will be presented from both the traditional and computational point of view. Music will be evaluated in three domains: visual, symbolic, and acoustic; and for each of them: formats, programming languages, and machine learning tools will be studied. This course will give a basic introduction to music information retrieval and computational musicology by identify problems and solutions for different kinds of musical genres and styles.
Literatur: <ul style="list-style-type: none">• Meinard Müller: "<i>Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications.</i>" Springer, ISBN: 978-3-319-21944-8. 2015.• Björn Schuller: "<i>Intelligent Audio Analysis</i>", Signals and Communication Technology, Springer, ISBN: 978-3642368059, 2013.
Moduleil: Music Informatics (Tutorial) Lehrformen: Übung Sprache: Englisch SWS: 2
Prüfung Music Informatics (Exam) Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0284: Praktikum Mobile Application Development		5 ECTS/LP
Version 1.0.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Students learn to realise concepts and models of mobile application development on the Android platform. They learn how to acquire data and signals from different sensors and to implement algorithms of pattern recognition and data/signal analysis on mobile devices. After participation in the Praktikum, students know how to analyse and structure complex problems in the field and to select suitable and state-of-the-art approaches to their solution.</p> <p>Participants are trained in their analytical and conceptional skills as well as in practical programming skills to transfer their knowledge to a practical task. They learn how to make scientifically meaningful assessments of their system using appropriate methods. They will know the mindset and thinking of application development and software engineering. All knowledge obtained during the Praktikum is applied in practice-oriented tasks. Furthermore, after participation, students will be able to recognise important technical evolution in the field of sensors, mobile apps and related tools, and smart devices.</p> <p>Students will work in teams and organise their work and task distribution in an autonomous way. They will learn how to summarise, present and document results in a reasonable way.</p> <p>Key skills: Formal methods; Methods for software development and abstraction; Versioning tools; Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams, Understanding of team management; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality Awareness.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
Voraussetzungen:		
Programming skills in Java are required.		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Praktikum Mobile Application Development</p> <p>Lehrformen: Praktikum</p> <p>Sprache: Englisch</p> <p>SWS: 4</p>
<p>Inhalte:</p> <p>In the Praktikum Mobile Application Development, students will develop software for the Android platform in groups of 2-4 students.</p> <p>Students will get to know and to program several groups of Android devices, e.g., mobiles and wearables, and respective sensors. They will familiarise with the Android Studio IDE, the according toolchain and the code management tool Gi.</p> <p>Topics: Java, Android, App development, UI design, Audio Player, Map and Location, Camera control, Mobile devices, Intelligent signal processing, Wearables.</p>

Literatur:

The literature is going to be announced by the lecturers during the course.

Prüfung

Praktikum Mobile Application Development

Praktikum

Modul INF-0291: Praktikum Selbstlernende Systeme		8 ECTS/LP
Version 1.2.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Praktikum sind die Studierenden in der Lage, Problemstellungen (höherer Komplexität) in dem Bereich „maschinelle Lernverfahren“ zu verstehen und zu lösen. Sie können unterschiedliche Verfahren vergleichen und einordnen und diese eigenständig auf konkrete Beispiele aus der Praxis anwenden. Studenten können intelligente Systeme im Bezug auf die algorithmische Lösung bewerten und sind weiterhin mit Verfahren zur Leistungsevaluierung eines intelligenten Systems vertraut. Sie sind außerdem 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. Weiterhin können sie bedeutende technische Enticklungen im Bereich „maschinelles Lernen“ erkennen und einordnen.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> - Fertigkeit zur Analyse und Strukturierung komplexer Informatikproblemstellungen, Fertigkeiten zur Entwicklung und Umsetzung von Lösungsstrategien - Kompetenz zur Vernetzung unterschiedlicher Fachgebiete - Präsentation und Dokumentation (eigener) Ergebnisse - Analytische methodische Kompetenz - Fähigkeit produktiv und zielführend im Team zu arbeiten - Akribisches Arbeiten - Fachübergreifende Kenntnisse - Systematische Weiterentwicklung von Entwurfsmodellen - Zeitmanagement - Eigenständige Literaturrecherche zu angrenzenden Themen - Grundsätze guter wissenschaftlicher Praxis 		
<p>Arbeitsaufwand: Gesamt: 240 Std. 225 Std. Praktikum (Selbststudium) 15 Std. Teilnahme an Lehrveranstaltungen (Präsenzstudium)</p>		
<p>Voraussetzungen: Empfohlen: die erfolgreiche Teilnahme an der Vorlesung OC2, Programmiererfahrung, Teamfähigkeit Modul Organic Computing II (INF-0066) - empfohlen</p>		<p>ECTS/LP-Bedingungen: Bestehen der Modulprüfung</p>
<p>Angebotshäufigkeit: jedes Semester</p>	<p>Empfohlenes Fachsemester: ab dem 2.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 2</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p> <p>Modulteil: Praktikum Selbstlernende Systeme Lehrformen: Praktikum Dozenten: Prof. Dr. Jörg Hähner Sprache: Deutsch SWS: 2 ECTS/LP: 8</p>		

Inhalte:

In dem Praktikum „Selbstlernende Systeme“ sollen die Studenten unterschiedliche Methoden aus dem Bereich „machine learning“ kennenlernen und vor allem selber implementieren. In einem Einführungskurs werden die benötigten theoretischen Grundlagen gelegt und diese dann innerhalb von Kleingruppen an konkreten Beispielen aus der Praxis angewandt.

Literatur:

aktuelle wissenschaftliche Paper

Zugeordnete Lehrveranstaltungen:

Praktikum zu Selbstlernende Systeme (Praktikum)

In dem Praktikum „Selbstlernende Systeme“ sollen die Studenten unterschiedliche Methoden aus dem Bereich „machine learning“ kennenlernen und vor allem selber implementieren. In einem Einführungskurs werden die benötigten theoretischen Grundlagen gelegt und diese dann innerhalb von Kleingruppen an konkreten Beispielen aus der Praxis angewandt.

Prüfung

Praktikum Selbstlernende Systeme

Portfolioprüfung, Kombination aus Praktischer und Schriftlich-Mündlicher Prüfung

Beschreibung:

schriftliche Abgaben, Softwareabnahme, Abschlussvortrag

Modul INF-0293: Advanced Deep Learning <i>Advanced Deep Learning</i>		8 ECTS/LP
Version 1.0.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: Nach der Teilnahme an dem Praxismodul verfügen die Studierenden über detailliertes und aktuelles Fachwissen auf dem Gebiet des maschinellen Lernens, können bedeutsame technische Entwicklungen identifizieren und können eine komplette Pipeline zur multimodalen Datenverarbeitung mit tiefen neuronalen Netzen implementieren. Sie können Problemstellungen und Ergebnisse des Gebiets präzise beschreiben, diskutieren und gelernte Konzepte und Methoden auf ähnliche Problemstellungen im maschinellen Lernen anwenden. Darüber hinaus analysieren die Studierenden weiterführende Konzepte, Methoden, Verfahren, Techniken und Technologien aus dem Gebiet des maschinellen Lernens, um sie in Forschungsprojekten anzuwenden, diese auf aktuelle industriennahe Aufgabenstellungen zu übertragen und dort aktiv mitzuarbeiten. Die Studierenden lernen wissenschaftlich anspruchsvolle Themenstellungen im Gebiet des maschinellen Lernens auf andere Forschungsfragen zu übertragen und darauf aufbauend ein komplexes Projekt in Gruppenarbeit auszuarbeiten. Außerdem verfügen sie über die Team- und Kommunikationsfähigkeit, um Problemstellungen auf dem Gebiet zu erörtern, Fragen und Zwischenergebnisse zu diskutieren, beschreiben und zu präsentieren. Zusätzlich sind die Studierenden in der Lage detaillierte Experimente durchzuführen und Ergebnisse zu beurteilen, vergleichen und auf Plausibilität zu prüfen.</p> <p>Schlüsselqualifikationen: Fortgeschrittene mathematisch-formale Methodik; Umsetzen fachlicher Lösungskonzepte in Programme und Modelle; Methoden zur Entwicklung größerer Softwaresysteme, Konstruktion von Abstraktionen und Architekturen; Fachübergreifende Kenntnisse; Systematische Weiterentwicklung von Entwurfsmethoden; Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten; Verstehen von Teamprozessen; Fertigkeit der Zusammenarbeit in Teams; Fähigkeit zur Leitung von Teams; Kennenlernen der Abläufe und Prozesse im Anwendungsumfeld der Informatik; Fertigkeit zur Lösung von Problemen unter praxisnahen Randbedingungen; Selbstreflexion; verantwortliches Handeln vor dem Hintergrund von Unzulänglichkeit und widerstreitenden Interessen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 20 Std. Vorlesung (Präsenzstudium) 40 Std. Übung (Präsenzstudium)</p>		
<p>Voraussetzungen: Grundkenntnisse in maschinellem Lernen und maschinellem Sehen (Grundstudiums Vorlesungen Multimedia Grundlagen 1 + 2 und Master-Vorlesung "Multimedia 2 - Machine Learning and Computer Vision") Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia II: Machine Learning and Computer Vision (INF-0092) - empfohlen</p>		<p>ECTS/LP-Bedingungen: Bestehen der Portfolioprüfung</p>
<p>Angebotshäufigkeit: jedes Semester</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 6</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	

Modulteile
Modulteil: Advanced Deep Learning (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2
Inhalte: <ul style="list-style-type: none">• Deep Learning in general• Deep Convolutional Neural Networks• Transfer Learning• Recurrent Neural Networks / LSTM Networks• Natural Language Processing• Multimodal Fusion (Vision+Language)• Application: Image Captioning
Zugeordnete Lehrveranstaltungen: Advanced Deep Learning (Vorlesung)
Modulteil: Advanced Deep Learning (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 4
Zugeordnete Lehrveranstaltungen: Übung zu Advanced Deep Learning (Übung)
Prüfung Advanced Deep Learning Portfolioprüfung, Die Endnote setzt sich aus bewerteten Übungsblättern und einem bewerteten Teamprojekt zusammen.

Modul INF-0294: Speech Pathology		5 ECTS/LP
Version 1.0.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>Knowledge: The students learn concepts relating to signal processing, speech production, phonetics, speech and language pathology, speech analysis, feature extraction, denoising and information reduction as exemplified through the analysis of automated voice pathology detection. They further gain insight into machine learning principles, with a particular focus on deep learning solutions, as is needed to diagnose a range of different voice pathologies. They will learn about different problems and solutions in the analysis of a variety of speech, relevant in the context of health care and wellbeing.</p> <p>Skills: The students will be able to choose appropriate algorithms of signal processing and machine intelligence, further develop these, design new solutions, and apply these to the task of voice pathology detection. They will know how to analyse and structure complex problems in the field, to employ suitable approaches to solve them, and to transfer knowledge to similar tasks. After participation in the course, they will be able to implement approaches and models into programs. Students will be able to assess developed systems in a scientific way. Important technical evolution and novelties in the fields of speech analysis and medical machine learning will be recognised by them.</p> <p>Competences: The students are able to characterise, judge on the quality and suitability, and design suited algorithmic solutions for intelligent signal analysis with a focus on voice pathology detection. They are further able to present and document results in a reasonable and meaningful way.</p> <p>Key skills: Formal methods; Knowledge of advantages and disadvantages of different design alternatives; Systematical advancement of design tools; Ability to work in teams; Knowledge of workflows and processes; Ability to find solutions for practical problems; Ability to work autonomously; Quality awareness; Scientific working; Literature research.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std.</p> <p>60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</p> <p>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p>		
<p>Voraussetzungen:</p> <p>Knowledge of basic mathematic lectures should be present.</p>		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Speech Pathology (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Englisch</p> <p>SWS: 2</p>		

Inhalte:

The course "Pathological Speech" will give an introduction to models of speech production (e.g., source-filter models) with a focus on aspects that are relevant to pathologies and their recognition using automated methods of signal processing and machine learning. Moreover, students learn about robust feature extraction, modern methods of machine learning and machine intelligence, and implementation of such systems on devices

Topics: Speech production; Phonetics; Speech and language pathology; Signal processing; Natural language processing; Speech analysis; Feature extraction; Machine learning; Deep learning; Denoising; Information reduction; Healthcare.

Literatur:

- Björn Schuller, Anton Batliner: "Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing", Wiley, ISBN: 978-1119971368, 2013.
- Further literature is going to be announced during the lecture.

Zugeordnete Lehrveranstaltungen:

Speech Pathology (Vorlesung)

In this novel course, students will learn about human speech production and related pathologies. The course will introduce recent source-filter models and the principal concepts of signal processing, feature extraction, and information reduction. State-of-the-art machine learning techniques will be discussed as well as implications of practical applications of speech-related health care technologies.

Modulteil: Speech Pathology (Übung)

Lehrformen: Übung

Sprache: Englisch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Speech Pathology (Übung)

Prüfung

Speech Pathology

Klausur / Prüfungsdauer: 90 Minuten

Modul INF-0296: Praktikum Interactive Machine Learning		8 ECTS/LP
Version 1.0.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul sind die Studierenden mit Grundkonzepten des interaktiven maschinellen Lernens vertraut. Sie sind in der Lage, fachliche Lösungskonzepte in Modelle zu übersetzen und beherrschen die Auswahl und sichere Anwendung geeigneter Methoden. Darüber hinaus erhalten sie einen Einblick in aktuelle Arbeiten des Forschungsgebiets. Weiterhin werden Kompetenzen in den Bereichen Team- und Kommunikationsfähigkeit sowie Selbstorganisation durch die Bearbeitung von Projektaufgaben und deren eigenständige Projektplanung vermittelt. Im Rahmen dieser Projektaufgaben werden Kenntnisse der Denkweise und Sprache des Forschungsgebietes durch die Diskussion und Präsentation der Projektergebnisse gefördert.</p> <p>Schlüsselqualifikationen: Umsetzen fachlicher Lösungskonzepte in Programme und Modelle; Fachspezifische Vertiefung; Fachübergreifende Kenntnisse; Fertigkeit, mit geeigneten Methoden wissenschaftlich aussagekräftige Bewertungen anzustellen; Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen; Fertigkeit der Zusammenarbeit in Teams; Kenntnisse von praxisrelevanten Aufgabenstellungen; Kennenlernen der Abläufe und Prozesse im Anwendungsumfeld der Informatik; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen</p>		
<p>Arbeitsaufwand: Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
Voraussetzungen: Programmiererfahrung		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 6	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Praktikum Interactive Machine Learning Lehrformen: Praktikum Sprache: Deutsch SWS: 6</p>
<p>Zugeordnete Lehrveranstaltungen: Interactive Machine Learning (Praktikum) Dieser Kurs fokussiert sich auf "interaktives Maschinelles Lernen" (iML). Wir definieren iML als Maschinelles Lernen bei dem der Mensch als zentraler Teil, interaktiv an verschiedenen Stellen in den Lernprozesses eingreift. Er überwacht dabei die Ergebnisse der Maschine und liefert Eingaben und Korrekturen um den Lernprozess zu verbessern. Während traditionelle ML Systeme menschliche Eingaben auf das Bereitstellen von Annotationen beschränken, beschäftigt sich iML damit Interaktionsdesign Richtlinien für ML Systeme zu erschaffen, und neue Methoden zu entwickeln menschliche Expertise in den ML Prozess einfließen zu lassen. Dadurch werden Systeme transparenter und nachvollziehbarer, was sich mit dem aktuell aufstrebenden Forschungstrend der "explainable AI" (XAI) überschneidet. Der Kurs besteht aus mehreren praktischen Hands-on Projekten und einer Einführung in verschiedene Themen rund um das Thema "interactive Machine Learning". Grundkenntnisse in C/C++ oder Python sind erwünscht. ... (weiter siehe Digicampus)</p>

Prüfung

Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation

Projektarbeit

Modul INF-0297: Praktikum Prozessorbau		5 ECTS/LP
Version 1.1.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Die Studierenden erwerben Kompetenzen in den folgenden Bereichen auf einem fortgeschrittenen, praxisorientierten, aber wissenschaftlichen Niveau: Entwurfsprozess für digitale Schaltungen, Schaltungslogik und Gatter, physikalische Grundlagen der elektronischen Bauteile, Beschreibung von Hardware mit einer Hardwarebeschreibungssprache. Zunächst lernen die Studierenden die Verknüpfung logischer Gatter und bauen einen Halb- und einen Volladdierer. Sie verstehen den Entwicklungsprozess für digitale Schaltungen und wenden diesen direkt praktisch an, indem sie ihren eigenen RISC-V-Prozessor entwickeln. Sie modellieren und implementieren ihn selbstständig mit Hilfe der Hardwarebeschreibungssprache VHDL. Dazu lernen sie die Vor- und Nachteile von schematischer und textueller Hardware-Beschreibung kennen und können entscheiden, wann es sinnvoll ist, welche Variante einzusetzen. Darüber hinaus kombinieren sie synchrone und asynchrone Prozesse, um ein gutes Zusammenspiel der Komponenten ihres selbstgebauten Mikroprozessors zu erreichen. Schließlich beurteilen die Studierenden die Effizienz ihrer Implementierung anhand der erzielten Taktfrequenz und des benötigten Hardware-Aufwandes. In einer abschließenden Projektphase lernen sie, eine komplexe Aufgabenstellung zu planen, nach einem selbst entwickelten fundierten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zur verständlichen Darstellung und Dokumentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewusstsein, Akribie; Projektgebundene Arbeit und Zeitmanagement; Auswahl und sichere Anwendung geeigneter Methoden; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Selbstreflexion</p>		
<p>Bemerkung: Im WiSe 2018/2019 statt "INF-0150: Hardware-Entwurf"</p>		
<p>Arbeitsaufwand: Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p>		
<p>Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen Modul Prozessorarchitektur (INF-0147) - empfohlen</p>		
Angebotshäufigkeit: einmalig WS	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 4	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Praktikum Prozessorbau		
Lehrformen: Praktikum		
Sprache: Deutsch		
SWS: 4		
<p>Inhalte: Die Veranstaltung "Prozessorbau" 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 Prozessorphipeline veranschaulicht. Das Ergebnis ist ein in VHDL entwickelter lauffähiger Prozessor für ein FPGA-Prototypenboard</p>		

Literatur:

- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010
- John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011

Zugeordnete Lehrveranstaltungen:

Praktikum Prozessorbau (Praktikum)

Prüfung

Praktikum Prozessorbau

praktische Prüfung

Modul ZCS-2000: Softskills		2 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Claudia Lange-Hetmann		
<p>Lernziele/Kompetenzen:</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"> - 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. - 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. - 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 oder vertiefen Teilaspekte wie u.a. Kreativität, Innovationsfähigkeit mit innovativen Methoden. <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>Bemerkung:</p> <p><u>Anmeldungspflicht:</u> Für die Teilnahme an den Kursen ist eine Anmeldung über digicampus/ VV erforderlich. <u>Anmeldephase:</u> 10. Jan – 31. Jan (für das folgende SS) bzw. 10. Juli – 31. Juli (für das folgende WS). Die Kurse finden größtenteils ab Mitte März bis letzten Sa* im April (SS) bzw. ab Mitte Sep. bis letzten Sa* im Okt. *vor Vorlesungsbeginn statt und haben eine limitierte Teilnehmerzahl pro Semester.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 60 Std.</p> <p>20 Std. Seminar (Präsenzstudium)</p> <p>5 Std. Vorbereitung von Präsentationen (Selbststudium)</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>Voraussetzungen:</p> <p>keine</p>		<p>ECTS/LP-Bedingungen:</p> <p>aktive Übungsteilnahme im Kurs</p>
<p>Angebotshäufigkeit: jedes Semester</p>	<p>Empfohlenes Fachsemester:</p> <p>ab dem 2.</p>	<p>Minimale Dauer des Moduls:</p> <p>1 Semester</p>

SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
------------------	---	--

Moduleile
<p>Moduleil: Softskills</p> <p>Lehrformen: Kurs</p> <p>Sprache: Deutsch / Englisch</p> <p>SWS: 2</p> <p>ECTS/LP: 2</p>
<p>Inhalte:</p> <p>Zur Auswahl stehen nachfolgende Kurse/Themen:</p> <p><u>(1) Kommunikationskompetenz</u></p> <ul style="list-style-type: none"> - Kommunikationstraining - Rhetorik - Präsentation - strategische Gesprächsführung <p><u>(2) Sozialkompetenz</u></p> <ul style="list-style-type: none"> - Konfliktmanagement - Moderation & Teamleitung - Führungskompetenzen entwickeln - Führungskompetenzen 4.0 - Agiles Management <p><u>(3) Methodenkompetenz</u></p> <ul style="list-style-type: none"> - Zeit-/Selbst-/Changemanagement - Besprechungsmanagement - Innovationen entwickeln - Design Thinking - Projektmanagement - Projektbasiertes Unternehmertum - Unternehmerisches Denken - U-Planspiel <p>Weiterehin können auch Kompakt-Kurse gewählt werden, bei denen die Teilnehmer o.g. Fähigkeiten erlernen und eine Projektaufgabe im Team bearbeiten:</p> <p>Kompaktkurs Märkte für Menschen Kompaktkurs Entrepreneurship ACE Startup-Challenge</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 http://www.uni-augsburg.de/de/career-service/studierende/veranstaltungen_fakultaet/ bzw. im VV Anmeldesystem https://thi-vv.informatik.uni-augsburg.de/vv/</p>

Literatur:

- (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
- (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
- (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 & Yves Pigneur, 2010. Campus Verlag.

Zugeordnete Lehrveranstaltungen:

Kompaktkurs - Märkte für Menschen (Kurs)

Inhalte: In diesem Kompaktkurs werden Inhalte von gesellschaftlicher Relevanz interdisziplinär erschlossen. Hierbei werden die folgenden drei Themenbereiche kontrovers diskutiert : • Ernährungsgewohnheiten vs. globale Ernährungssicherheit • (Finanz-)Märkte vs. Gerechtigkeit • Leistungsgesellschaft vs. sozialer Systemgedanke Jeder dieser Themenbereiche wird zunächst mit Hilfe wissenschaftlicher Inputs interdisziplinär erschlossen. Daran anschließend erarbeitet sich jede Kleingruppe fundiertes Wissen zu ihrem Themenbereich, setzt sich differenziert mit kontroversen Perspektiven auseinander und präsentiert die Ergebnisse anschließend kontrovers in einer „Talk-Show“ bzw. Podiumsdiskussion. Die Studierenden erarbeiten in Projektgruppen selbst konkrete Vorschläge und Projekte zur Gestaltung einer besseren Zukunft im Rahmen der obigen Themenkomplexe und präsentieren diese zum Abschluss des Seminars. Parallel zu dieser inhaltlichen Arbeit werden methodische Kenntnisse und Hintergrundinf ... (weiter siehe Digicampus)

Soft-Skill-Kurse WS 2018/19 für MPing_I (Kurs)

Startup Challenge - Augsburg Center for Entrepreneurship (Bachelorseminar)

• Innovatives Entrepreneurship Format und Startup Projektstudium • Anhand einer realen unternehmerischen Entscheidungssituation können die Studierenden ihr ganzes unternehmerisches Talent testen • Die Studierenden lernen wesentliche Methoden und Konzepte zur Steuerung und Bewertung von Geschäftsmodellen, Pricing Strategien, Vertrieb und Marketing. • Experten und renommierte Startup-Mentoren begleiten die Studierenden in fachlichen Fragen. • Fachvorträge zu ausgewählten Themenstellungen runden das Seminarangebot ab; sowie die Möglichkeit mit der Augsburger Startup Szene auf Tuchfühlung zu gehen.

Startup Challenge - Augsburg Center for Entrepreneurship (Masterseminar) (Seminar)

• Innovatives Entrepreneurship Format und Startup Projektseminar • Anhand einer realen unternehmerischen Entscheidungssituation können die Studierenden ihr ganzes unternehmerisches Talent testen • Die Studierenden lernen wesentliche Methoden und Konzepte zur Steuerung und Bewertung von Geschäftsmodellen, Pricing Strategien, Vertrieb und Marketing. • Experten und renommierte Startup-Mentoren begleiten die Studierenden in fachlichen Fragen. • Fachvorträge zu ausgewählten Themenstellungen runden das Seminarangebot ab; sowie die Möglichkeit mit der Augsburger Startup Szene auf Tuchfühlung zu gehen.

Prüfung

Anwesenheit und aktive Übungsteilnahme im Kurs

Beteiligungsnachweis, unbenotet

Modul INF-0039: Seminar über Software Engineering verteilter Systeme (MA)		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet Software Engineering für verteilte 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 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.</p> <p>Schlüsselqualifikation: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Seminar über Software Engineering verteilter Systeme Lehrformen: Seminar Sprache: Deutsch SWS: 2</p>		
Inhalte: Aktuelle Software Engineering-Themen aus Industrie und Forschung.		
Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.		
Zugeordnete Lehrveranstaltungen:		

Seminar über Software Engineering verteilter Systeme f. Master (Seminar)

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.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0040: Seminar Grundlagen des Software Engineering für Automotive Systems (MA)		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet Automotive Software Engineering 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.</p> <p>Schlüsselqualifikation: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<p>Voraussetzungen: keine</p>		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Seminar Grundlagen des Software Engineering für Automotive Systems Lehrformen: Seminar Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: Dieses Seminar soll die Grundlagen des Systems & Software Engineering im Automotive Bereich behandeln. Es werden dabei Aspekte der Vorlesung Automotive Software Engineering aufgenommen und vertieft.</p>		
<p>Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.</p>		
Zugeordnete Lehrveranstaltungen:		

Seminar zu Automotive Software Engineering (Master) (Seminar)

Bestandteil dieses Seminars sind Ansätze und Techniken im Bereich des Automotive Software Engineerings. 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.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0041: Seminar Grundlagen des Software Engineering für Avionic Systems (MA)		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet Avionic Software Engineering 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.</p> <p>Schlüsselqualifikation: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Seminar Grundlagen des Software Engineering für Avionic Systems		
Lehrformen: Seminar		
Sprache: Deutsch		
SWS: 2		
Inhalte: 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.		
Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.		
Zugeordnete Lehrveranstaltungen:		

Seminar zu Avionic Software Engineering (Master) (Seminar)

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.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0058: Seminar Algorithmen und Datenstrukturen für Master		4 ECTS/LP
Version 1.5.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
<p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, sich fachliche Inhalte der Algorithmik eigenständig aus anspruchsvoller wissenschaftlicher Originalliteratur zu erarbeiten und diese kritisch zu bewerten. Sie können das Gelesene in einen größeren Kontext einordnen, eine sinnvolle Themenauswahl treffen und die gewählten fachlichen Inhalte klar und verständlich schriftlich darstellen und mündlich frei vortragen. Sie verstehen es, einen Vortrag gut zu strukturieren, interessant und motivierend zu gestalten und an die Voraussetzungen der Zuhörer und den vorgegebenen zeitlichen Rahmen anzupassen.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken, zur kritischen Stellungnahme und zur überzeugenden fachlichen Argumentation; Fähigkeit zur Literaturrecherche und zum eigenständigen Arbeiten mit englischsprachiger Fachliteratur; Fähigkeit zur Bewertung von Lösungsansätzen, Verfahren und Techniken unter unterschiedlichen Gesichtspunkten; Qualitätsbewusstsein; Akribie; Kommunikationsfähigkeit; Zeitmanagement.</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen: Empfehlenswert: Gutes Verständnis des Informatik III-Stoffes.		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Moduleile		
Modulteil: Seminar Algorithmen und Datenstrukturen		
Lehrformen: Seminar		
Sprache: Deutsch		
SWS: 2		
Inhalte: Aktuelle und klassische Themen aus dem Bereich Algorithmen und Datenstrukturen werden anhand von Originalliteratur behandelt.		
Literatur: Ausgewählte wissenschaftliche Artikel.		
Prüfung		
Schriftliche Ausarbeitung und Vortrag Seminar		

Modul INF-0070: Seminar Organic Computing		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen:</p> <p>Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet Organic Computing selbstständig zu erarbeiten, zu analysieren und bezogen auf das individuelle Seminarthema aus dem genannten Gebiet zu bewerten.</p> <p>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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen.</p> <p>Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an.</p> <p>Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.</p> <p>Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewusstsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 120 Std.</p> <p>90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p> <p>30 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen:		
keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Moduleile		
Modulteil: Seminar Organic Computing		
Lehrformen: Seminar		
Sprache: Deutsch		
SWS: 2		
Inhalte:		
Die Themen des Seminars werden jedes Jahr neu festgelegt und aktuellen Trends angepasst.		
Literatur:		
Literatur in Abhängigkeit von den aktuellen Themen: wiss. Paper oder Bücher		
Zugeordnete Lehrveranstaltungen:		

Seminar über Organic Computing (Seminar)

http://www.informatik.uni-augsburg.de/lehrstuehle/oc/lehre/WS_1819/S-OC/

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0071: Seminar Naturalogische Algorithmen und Multiagentensysteme		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen:</p> <p>Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf den Gebieten der naturinspirierten Algorithmen sowie den Multiagentensystemen selbstständig zu erarbeiten, zu analysieren und bezogen auf das individuelle Seminarthema aus dem genannten Gebiet zu bewerten.</p> <p>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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen.</p> <p>Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an.</p> <p>Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.</p> <p>Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewusstsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 120 Std.</p> <p>30 Std. Seminar (Präsenzstudium)</p> <p>90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
Voraussetzungen:		
keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Seminar Naturalogische Algorithmen und Multiagentensysteme</p> <p>Lehrformen: Seminar</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p>		

Inhalte:

In dem Seminar sollen sich die Studenten jeweils einem speziellen Thema in Schnittbereich naturanaloge 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.

Literatur:

wird im Seminar bekanntgegeben

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0084: Seminar Next Generation Networks		4 ECTS/LP
Version 1.0.0 (seit SoSe15) Modulverantwortliche/r: Prof. Dr. Rudi Knorr		
<p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, ein wissenschaftlich anspruchsvolles Thema aus dem Gebiet "Next Generation Networks" selbständig zu erarbeiten, zu analysieren und 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. Außerdem können sie die Strukturen des logischen Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.</p> <p>Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten; Qualitätsbewusstsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement.</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
Voraussetzungen: Modul Kommunikationssysteme (INF-0081) - empfohlen		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Moduleile		
Modulteil: Seminar Next Generation Networks		
Lehrformen: Seminar		
Sprache: Deutsch		
SWS: 2		
Inhalte: Die Themen für dieses Seminar werden jedes Jahr unter Berücksichtigung aktueller Trends aus dem Gebiet "Next Generation Networks" neu festgelegt.		
Literatur: Grundlegende und aktuelle Forschungsliteratur in Abhängigkeit von den festgelegten Themen.		

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0095: Seminar Multimedia Computing (MA)		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: 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 maschinellen Sehens (z.B. Bildverarbeitung, Videoverarbeitung, maschinelles Sehen/Hören und Lernen, Bild-/Videosuche) selbstständig zu erarbeiten, zu analysieren und bezogen auf das individuelle Seminarthema aus dem genannten Gebiet zu bewerten. Sie verfügen über 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.</p> <p>Schlüsselqualifikationen: Erlernen von Präsentationstechniken, Literaturrecherche; Arbeit mit englischer Fachliteratur; Grundsätze guter wissenschaftlicher Praxis; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen:		
keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Seminar Multimedia Computing (MA)		
Lehrformen: Seminar		
Sprache: Deutsch		
SWS: 2		
Inhalte: Das konkrete Thema des Seminars aus dem weitläufigen Gebiet des Multimedia wird jedes Jahr neu festgelegt und an aktuelle Themen angepasst.		
Literatur: aktuelle Forschungsliteratur		
Prüfung		
Vortrag und schriftliche Ausarbeitung Seminar		

Modul INF-0118: Seminar Programmiermethodik und Multimediale Informationssysteme für Master		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p>Lernziele/Kompetenzen: 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>Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten.</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: in der Regel mind. 1x pro Studienjahr	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Seminar Programmiermethodik und Multimediale Informationssysteme für Master Lehrformen: Seminar Sprache: Deutsch SWS: 2</p>
<p>Inhalte: Themen aus den Bereichen "Theoretische Informatik", "Multimedia" oder "Datenbanken und Informationssysteme"</p>
<p>Literatur: wird jeweils bekanntgegeben</p>
<p>Zugeordnete Lehrveranstaltungen: Seminar über Theoretische Informatik (Seminar) Im Rahmen des Seminars werden Themen aus dem Bereich der formalen Methoden, insbes. der algebraischen Semantik/algebraischen Logik behandelt. Das Seminar findet als Blockveranstaltung statt. Die Themen (-bereiche) werden zu Beginn der Vorlesungszeit hier veröffentlicht. Die Zuteilung erfolgt in einer ersten Besprechung vor Ort. In der Semestermitte (Ende Dezember/Anfang Januar) findet ein Zwischenbesprechungstermin. Die Präsentation findet in der vorlesungsfreien Zeit statt. Der genaue Termin wird bei der Zwischenbesprechung</p>

festgelegt. Präsentationsmaterialien müssen 1 Woche vor dem Vortragstermin eingereicht werden. Die schriftliche Ausarbeitung muß 2 Wochen nach dem Vortragstermin eingereicht werden. Die Note berechnet sich 50/50 aus Vortrag und Ausarbeitung.
... (weiter siehe Digicampus)

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0136: Seminar Software- und Systems Engineering (Master)		4 ECTS/LP
Version 1.1.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen:</p> <p>Nach dem erfolgreichen Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien aus dem Gebiet des Software und Systems Engineering selbstständig zu erarbeiten, zu analysieren und bezogen auf das individuelle Seminarthema aus dem genannten Gebiet zu bewerten.</p> <p>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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen.</p> <p>Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an.</p> <p>Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Literaturrecherche • Eigenständiges Arbeiten mit auch englischsprachiger Fachliteratur • Analytisch-methodische Kompetenz • Wissenschaftliche Methodik • Grundsätze guter wissenschaftlicher Praxis • Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation • Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und präziser Argumentation • Qualitätsbewußtsein, Akribie • Kommunikationsfähigkeit • Zeitmanagement • Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten 		
<p>Arbeitsaufwand:</p> <p>Gesamt: 120 Std.</p> <p>90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p> <p>30 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	

Modulteile
<p>Modulteil: Seminar Software- und Systems Engineering (Master) Lehrformen: Seminar Sprache: Deutsch Angebotshäufigkeit: jedes Sommersemester SWS: 2</p>
<p>Inhalte: 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.</p>
<p>Literatur: Abhängig von den konkreten Themen des Seminars</p>
<p>Zugeordnete Lehrveranstaltungen:</p> <p>Seminar zu Software- und Systems Engineering (Master) (Seminar) In diesem Semester beschäftigt sich das Seminar zu Software und Systems Engineering mit der Thematik von Machine Learning in der Robotik. Die Studierenden sollen jeweils zu einem vorgegebenen Unterthemenbereich (z. B. Reinforcement Learning, Transfer Learning, Erlernen von Schwarmverhalten, Signalverarbeitung, o.ä.) zuerst eine Themenstellung auswählen und konkretisieren, und diese dann in Form einer Ausarbeitung und eines Posters darstellen. Voraussichtlicher Ablauf: - Kick-Off-Veranstaltung - Eigenständiger Literaturüberblick und Themenkonkretisierung - Kurzvorstellung des gewählten Themas, Feedback - Ausarbeitung des Themas und Erstellung eines Posters - Abschlusspräsentation mit kurzem Pitch-Vortrag und Besprechung der Poster Die Abschlusspräsentation findet am Ende der Vorlesungszeit oder zu Beginn der Vorlesungszeit als Blockveranstaltung an 1 oder 2 Terminen statt. Die genauen Termine werden im Rahmen des Seminars festgelegt. ... (weiter siehe Digicampus)</p>
<p>Prüfung Seminar Software- und Systems Engineering (Master) Seminar / Prüfungsdauer: 45 Minuten Bearbeitungsfrist: 3 Monate</p>

Modul INF-0152: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden. Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Sommersemester	Empfohlenes Fachsemester: 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen		
Lehrformen: Seminar		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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>		

Literatur:

individuell gegeben und Selbstrecherche

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0153: Seminar Safety-Critical Systems		4 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: 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 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden. Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Wintersemester	Empfohlenes Fachsemester: 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Seminar Safety-Critical Systems		
Lehrformen: Seminar		
Sprache: Deutsch		
SWS: 2		
<p>Inhalte: 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>		

Literatur:

individuell gegeben und Selbstrecherche

Zugeordnete Lehrveranstaltungen:

Seminar Safety-Critical Systems (Master) (Seminar)

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.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0164: Seminar Theorie verteilter Systeme A		4 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "Theorie verteilter 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 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden. Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von theoretischen Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: unregelmäßig	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Seminar Theorie verteilter Systeme A		
Lehrformen: Seminar		
Sprache: Deutsch		
SWS: 2		
Inhalte: Es werden Arbeiten zu verschiedenen Themen aus dem Bereich "Theorie verteilter Systeme" behandelt.		
Literatur: wird jeweils bekanntgegeben		

Prüfung

Schriftliche Ausarbeitung

Seminar

Modul INF-0227: Seminar Datenbanksysteme für Master		4 ECTS/LP
Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Peter Michael Fischer		
<p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet der Datenbanksysteme 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden. Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<p>Voraussetzungen: Modul Datenbanksysteme (INF-0073) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 2</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Seminar Datenbanksysteme für Master Lehrformen: Seminar Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: Aktuelle Forschungsbeiträge aus den Bereich "Datenbanken und Informationssysteme".</p>		
<p>Literatur: Aktuelle Forschungsbeiträge</p>		

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0232: Seminar Medical Information Sciences (MA)		4 ECTS/LP
Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet Medical Information Sciences 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.</p> <p>Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
<p>Voraussetzungen: keine</p>		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Medical Information Sciences (Seminar) Lehrformen: Seminar Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: 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.</p>		
<p>Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.</p>		
Zugeordnete Lehrveranstaltungen:		

Seminar zu Medical Information Sciences f. Master (Seminar)

Bestandteil dieses Seminars sind fortgeschrittene Ansätze und Techniken im Bereich Medical Information Science.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0240: Seminar Informationssysteme für Master		4 ECTS/LP
Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Peter Michael Fischer		
<p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet der Informationssysteme 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden. Schlüsselqualifikationen: Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</p>		
<p>Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<p>Voraussetzungen: Modul Datenbanksysteme (INF-0073) - empfohlen</p>		
<p>Angebotshäufigkeit: unregelmäßig (i. d. R. im WS)</p>	<p>Empfohlenes Fachsemester: ab dem 1.</p>	<p>Minimale Dauer des Moduls: 1 Semester</p>
<p>SWS: 2</p>	<p>Wiederholbarkeit: siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Seminar Informationssysteme für Master Lehrformen: Seminar Sprache: Deutsch SWS: 2</p>		
<p>Inhalte: Aktuelle Forschungsbeiträge aus den Bereich "Datenbanken und Informationssysteme".</p>		
<p>Literatur: Aktuelle Forschungsbeiträge</p>		
<p>Zugeordnete Lehrveranstaltungen: Seminar Informationssysteme für Master (Seminar)</p>		

Veranstaltungen wie SIGMOD, ICDE oder VLDB gehören zu den renommiertesten internationalen Konferenzen im Bereich Datenbanksysteme. Hier werden in Sessions und Workshops von internationalen Forscherteams aktuelle Ergebnisse zu einem breiten Themenspektrum von Streamverarbeitung, Big Data, Sozialen Netzwerken, Graphdatenbanken bis hin zu Concurrency und Hardwareunterstützung präsentiert. Im Zuge des Seminars sollen ausgewählte Beiträge dieser Konferenzen vorgestellt und in einem größeren Kontext eingebettet werden. Zudem sollen die Grundlagen wissenschaftlichen Arbeitens näher betrachtet werden.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0251: Seminar Artificial Intelligence		4 ECTS/LP
Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: PD Dr. Jonghwa Kim		
Inhalte: 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.		
Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "Künstliche Intelligenz und Intelligente 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 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. Außerdem können sie die logischen Strukturen des Denkens und Argumentierens erkennen und zielführend einsetzen. Die Teilnehmenden können klar und verständlich formulieren und Fachinhalte frei vortragen. Sie verstehen es, einen Vortrag klar und nachvollziehbar zu strukturieren und auch bei komplexen Inhalten den Vortrag auf wesentliche Botschaften auszurichten und diese verständlich zu vermitteln. Argumentationsketten und Lösungsstrategien bei Störungen wenden sie gekonnt an. Die Studierenden verstehen es, präsent aufzutreten und souverän mit gängigen Präsentationsmedien umzugehen und diese interaktiv einzusetzen. Sie schaffen es, einen Vortrag auf eine bestimmte Zielgruppe auszurichten und den Zuhörer auch bei längeren Vortragsdauern zu motivieren und verschiedene Moderationstechniken anzuwenden.		
Schlüsselqualifikationen: Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten; Literaturrecherche; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; Analytisch-methodische Kompetenz; Wissenschaftliche Methodik; Grundsätze guter wissenschaftlicher Praxis; Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und zu deren Dokumentation; Fertigkeit zum logischen, abstrakten, analytischen und konzeptionellen Denken und formaler Argumentation; Qualitätsbewußtsein, Akribie; Kommunikationsfähigkeit; Zeitmanagement		
Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)		
Voraussetzungen: keine		ECTS/LP-Bedingungen: Bestehen der Modulprüfung
Angebotshäufigkeit: in der Regel mind. 1x pro Studienjahr	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Seminar Artificial Intelligence Lehrformen: Seminar Sprache: Deutsch / Englisch SWS: 2 ECTS/LP: 4		

Inhalte:

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.

Literatur:

aktuelle Forschungsliteratur

Zugeordnete Lehrveranstaltungen:

Seminar Artificial Intelligence (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

Modul INF-0274: Seminar Embedded Intelligence for Health Care and Wellbeing (Master)		4 ECTS/LP
Version 1.0.0 (seit WS17/18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>After attending the seminar, students will be able to autonomously acquire and understand advanced problem statements, concepts, methods, approaches, techniques, and technologies in the field of Embedded Intelligence for Health Care and Wellbeing. They possess the scientific techniques, communication skills, and the ability to employ suitable media, to present understandingly a special topic in spoken and written, and to discuss and evaluate scientifically challenging themes from the field in a critical way. Furthermore, they can recognise logical structures of thinking and debating and employ them constructively.</p> <p>Participants can express themselves in a clear and understandable way and present scientific topics. They understand how to structure a talk, to focus it - also given a complex content - on the essential messages, and to communicate them in a suitable way. The lines of arguments and strategies in case of disturbances are applied by the students. Students know how to perform energetically, to cope with the presentation media and to use them interactively. They manage to orient a talk toward a certain audience, to motivate the listeners also over a longer duration, and to employ different methods of moderation.</p> <p>Key skills: Fundamentals of good scientific practice; Analytical-methodological competency; Time management; Literature research; Self-contained work with English technical literature; Communication skills; Ability to present (in written and spoken) practical and theoretical ideas in an understandable, confident, and convincing way; Writing a report in the markup language LaTeX; Evaluation of methods, technologies, and solutions w.r.t. different aspects; Quality awareness.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen:		
keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 5.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Seminar Embedded Intelligence for Health Care and Wellbeing (Master)		
Lehrformen: Seminar		
Sprache: Deutsch / Englisch		
SWS: 2		
<p>Inhalte:</p> <p>In the seminar Embedded Intelligence for Health Care and Wellbeing, recent research works in this field are going to be discussed. This comprises both the acquisition of data through sensors and (e.g., microphones or electrodes) and the analysis and the modelling of the data. One important aspect is also the practicability of modern deep learning methods. Health Care and Wellbeing applications reach from tracking of health states (e.g., epilepsy or depression) to personal assistance services.</p> <p>The participating students will work on a certain aspect, supervised by a research associate of the chair. They will summarise their results in a written report and an oral presentation.</p> <p>Topics: E-Health, M-Health, Sensor Signal Analysis, Vital Signs, Big Data.</p>		

Literatur:

Wird vom Dozenten oder der Dozentin bekannt gegeben

Zugeordnete Lehrveranstaltungen:

Seminar Embedded Intelligence for Health Care and Wellbeing (Bachelor & Master) (Seminar)

The specific topics are introduced and assigned to students in the kick-off meeting during the first week of the lecture period. During the first meetings, a tutorial on oral presentation techniques and literature research is given. Students are going to follow on their topics under the supervision of the scientific staff of the chair. By the end of the lecture period, a short oral presentation has to be given by each student. A written report (survey paper) must be handed in afterwards.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0281: Seminar Sports Informatics (Master)		4 ECTS/LP
Version 1.0.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>After attending the seminar, students will be able to autonomously acquire and understand advanced problem statements, concepts, methods, approaches, techniques, and technologies in the field of Sports Informatics. They possess the scientific techniques, communication skills, and the ability to employ suitable media, to present understandingly a special topic in spoken and written, and to discuss and evaluate scientifically challenging themes from the field in a critical way. Furthermore, they can recognise logical structures of thinking and debating and employ them constructively.</p> <p>Participants can express themselves in a clear and understandable way and present scientific topics. They understand how to structure a talk, to focus it - also given a complex content - on the essential messages, and to communicate them in a suitable way. The lines of arguments and strategies in case of disturbances are applied by the students. Students know how to perform energetically, to cope with the presentation media and to use them interactively. They manage to orient a talk toward a certain audience, to motivate the listeners also over a longer duration, and to employ different methods of moderation.</p> <p>Key skills: Fundamentals of good scientific practice; Analytical-methodological competency; Time management; Literature research; Self-contained work with English technical literature; Communication skills; Ability to present (in written and spoken) practical and theoretical ideas in an understandable, confident, and convincing way; Writing a report in the markup language LaTeX; Evaluation of methods, technologies, and solutions w.r.t. different aspects; Quality awareness.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
Voraussetzungen:		
keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Seminar Sports Informatics (Master)		
Lehrformen: Seminar		
Sprache: Deutsch / Englisch		
SWS: 2		
<p>Inhalte:</p> <p>In the seminar Sports Informatics, recent research works in this field are going to be discussed. Research in Sports Informatics comprises both the acquisition of data through sensors and wearables (e.g., acceleration/skin conductance/pulse sensors) and the analysis and the modelling of the data. One important aspect is also the practicability of modern deep learning methods. Applications of Sports Informatics reach from tracking of fitness parameters to improving training methods of professional athletes.</p> <p>The participating students will work on a certain aspect, supervised by a research associate of the chair. They will summarise their results in a written report and an oral presentation.</p> <p>Topics: Sensors, Wearables, Signal analysis, Machine Learning, Health Care.</p>		

Literatur:

Wird vom Dozenten / von der Dozentin bekannt gegeben

Zugeordnete Lehrveranstaltungen:

Seminar Sports Informatics (Bachelor & Master) (Seminar)

The specific topics are introduced and assigned to students in the kick-off meeting during the first week of the lecture period. During the first meetings, a tutorial on oral presentation techniques and literature research is given. Students are going to follow on their topics under the supervision of the scientific staff of the chair. By the end of the lecture period, a short oral presentation has to be given by each student. A written report (survey paper) must be handed in afterwards.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0283: Seminar Computer Audition (Master)		4 ECTS/LP
Version 1.1.0 (seit SoSe18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen:</p> <p>After attending the seminar, students will be able to autonomously acquire and understand advanced problem statements, concepts, methods, approaches, techniques, and technologies in the field of Computer Audition. They possess the scientific techniques, communication skills, and the ability to employ suitable media, to present understandingly a special topic in spoken and written, and to discuss and evaluate scientifically challenging themes from the field in a critical way. Furthermore, they can recognise logical structures of thinking and debating and employ them constructively.</p> <p>Participants can express themselves in a clear and understandable way and present scientific topics. They understand how to structure a talk, to focus it - also given a complex content - on the essential messages, and to communicate them in a suitable way. The lines of arguments and strategies in case of disturbances are applied by the students. Students know how to perform energetically, to cope with the presentation media and to use them interactively. They manage to orient a talk toward a certain audience, to motivate the listeners also over a longer duration, and to employ different methods of moderation.</p> <p>Key skills: Fundamentals of good scientific practice; Analytical-methodological competency; Time management; Literature research; Self-contained work with English technical literature; Communication skills; Ability to present (in written and spoken) practical and theoretical ideas in an understandable, confident, and convincing way; Writing a report in the markup language LaTeX; Evaluation of methods, technologies, and solutions w.r.t. different aspects; Quality awareness.</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
Voraussetzungen:		
keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Seminar Computer Audition (Master)		
Lehrformen: Seminar		
Sprache: Deutsch / Englisch		
SWS: 2		
<p>Inhalte:</p> <p>In the seminar Computer Audition, topics of machine learning and signal analysis related to audio signals are discussed. Audio data comprises here different classes of signals, i.e., speech, music, or general sound. Applications of the discussed algorithms can be different: automatic speech recognition, acoustic scene classification, or music information retrieval. Especially modern architectures and methods of Deep Learning play an important role.</p> <p>The participating students will work on a certain aspect, supervised by a research associate of the chair. They will summarise their results in a written report and an oral presentation.</p> <p>Topics: Speech recognition, Computational paralinguistics, Music Information Retrieval, Deep Learning.</p>		

Literatur:

To be announced by the lecturers.

Zugeordnete Lehrveranstaltungen:

Seminar Computer Audition (Bachelor & Master) (Seminar)

The specific topics are introduced and assigned to students in the kick-off meeting during the first week of the lecture period. During the first meetings, a tutorial on oral presentation techniques and literature research is given. Students are going to follow on their topics under the supervision of the scientific staff of the chair. By the end of the lecture period, a short oral presentation has to be given by each student. A written report (survey paper) must be handed in afterwards.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

Modul INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme		10 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet Softwaremethodiken für verteilte Systeme und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; 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>Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Projektmodul Softwaremethodiken für verteilte Systeme		
Lehrformen: Praktikum		
Sprache: Deutsch		
SWS: 1		
Inhalte: Aktuelle Forschungsthemen an der Professur Softwaremethodik für verteilte Systeme		
Literatur: Wird zu den jeweiligen Themen bereitgestellt.		
Zugeordnete Lehrveranstaltungen: Oberseminar zu Softwaremethodik für verteilte Systeme		
Prüfung Projektabnahme, Vortrag, Abschlussbericht Praktikum		

Modul INF-0059: Projektmodul Theoretische Informatik		10 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Torben Hagerup		
Lernziele/Kompetenzen: Lernziele: Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet der theoretischen Informatik zu verstehen, und sie verfügen dort über tiefergehende Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren und Techniken der theoretischen Informatik in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung dabei auftretender fachlicher Probleme einzusetzen. Dadurch haben sie die nötigen Voraussetzungen, um eigene wissenschaftliche Leistungen zu erbringen und an die internationale Forschung anzuknüpfen. Darüber hinaus verfügen die Studierenden über die Kommunikationsfähigkeit, die Fähigkeit zur eigenständigen Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen der theoretischen Informatik zu diskutieren, Zwischenziele zu definieren, Lösungsvorschläge kritisch zu bewerten und eigene Ideen einzuordnen und zu präsentieren. Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Qualitätsbewusstsein, Akribie, Selbständige Arbeit, Zeitmanagement, eigenständige Literaturrecherche, Arbeit mit englischsprachiger Fachliteratur, Grundsätze guter wissenschaftlicher Praxis.		
Arbeitsaufwand: Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	

Moduleile
Modulteil: Projektmodul Theoretische Informatik Lehrformen: Praktikum Sprache: Deutsch SWS: 1
Inhalte: Autonome Mitarbeit an aktuellen Forschungsthemen.
Zugeordnete Lehrveranstaltungen: Oberseminar Theoretische Informatik

Prüfung Vortrag und schriftliche Ausarbeitung Praktikum

Modul INF-0072: Projektmodul Organic Computing		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet "Organic Computing" und verfügen dort über tiefergehende Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; 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>Arbeitsaufwand: Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Projektmodul Organic Computing		
Lehrformen: Praktikum		
Sprache: Deutsch		
SWS: 1		
Inhalte: Mitarbeit an aktuellen Forschungsthemen.		
<p>Literatur: In Abhängigkeit vom zu bearbeitenden Thema:</p> <ul style="list-style-type: none"> • Paper • Buch • Handbuch 		
Zugeordnete Lehrveranstaltungen:		
Oberseminar Organic Computing Weiterführende Themen aus dem Bereich Organic Computing		

Prüfung

Vortrag und Abschlussbericht

Praktikum

Modul INF-0080: Projektmodul Datenbanken und Informationssysteme		10 ECTS/LP
Version 1.2.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Peter Michael Fischer		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet der Datenbanken und Informationssysteme und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; 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>Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
<p>Voraussetzungen: Modul Datenbanksysteme (INF-0073) - empfohlen Modul Suchmaschinen (INF-0077) - empfohlen</p>		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Projektmodul Datenbanken und Informationssysteme Lehrformen: Praktikum Sprache: Deutsch SWS: 1</p>		
<p>Inhalte: Arbeiten am Präferenz-SQL-System des Lehrstuhls</p>		
<p>Literatur:</p> <ul style="list-style-type: none"> • Aktuelle Forschungsbeiträge zum Thema "Präferenzen" • Handbücher 		
<p>Zugeordnete Lehrveranstaltungen: Oberseminar Datenbanken und Informationssysteme</p>		
<p>Prüfung Softwareabnahme, Vortrag, Abschlussbericht Praktikum</p>		

Modul INF-0085: Projektmodul Kommunikationssysteme		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rudi Knorr		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet der Kommunikationssysteme und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständiges Arbeiten mit englischsprachiger Fachliteratur; 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>Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Projektmodul Kommunikationssysteme		
Lehrformen: Praktikum		
Sprache: Deutsch		
SWS: 1		
Inhalte: Aktuelle Forschungsthemen auf dem Gebiet "Kommunikationssysteme".		
Literatur: wissenschaftliche Papiere, Handbücher		
Zugeordnete Lehrveranstaltungen:		
Oberseminar Systemnahe Informatik und Kommunikationssysteme		
Es werden aktuelle Themen aus dem Bereich der Systemnahen Informatik und Kommunikationssysteme vorgetragen und diskutiert. Dazu zählen bei Lst. Knorr Selbstorganisierende Kommunikationssysteme und Next Generation Networks (NGN) und bei Lst. Ungerer Prozessorarchitekturen und Parallelisierung für harte Echtzeitsysteme, Fehlertoleranz und Transaktionsspeicher.		

Prüfung

Vortrag und Abschlussbericht

Praktikum

Modul INF-0096: Projektmodul Multimedia Computing		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf 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) und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; 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>Arbeitsaufwand: Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Projektmodul Multimedia Computing		
Lehrformen: Praktikum		
Sprache: Deutsch		
Angebotshäufigkeit: nach Bedarf		
SWS: 1		
Inhalte: 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.		
Literatur: Literaturhinweise werden zum Beginn der Veranstaltung bekanntgegeben.		
Zugeordnete Lehrveranstaltungen: Oberseminar Multimedia Computing		

Prüfung

**Vortrag mit Softwarerepräsentation; Ausarbeitung mit Softwaredokumentation; Erklärung des Quellcodes
(Code Review)**

Praktikum

Modul INF-0108: Projektmodul Lehrprofessur für Informatik <i>Project Module Teaching Professorship Informatics</i>		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Robert Lorenz		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf einem der Gebiete "Nebenläufige Systeme", "Petrietze" oder "Process Mining" und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; 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>Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
Voraussetzungen: Grundkenntnisse in einschlägigen Forschungsgebieten des Lehrstuhls		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Projektmodul Lehrprofessur für Informatik Lehrformen: Praktikum Sprache: Deutsch / Englisch SWS: 1</p>		
<p>Inhalte: 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 Petrietzen aus nicht-sequentiellen Verhaltensbeschreibungen, Process Mining Techniken, Entfaltung von Petrietzen und Entfaltungs-basiertes Model-Checking, Finite State Transducer in der semantischen Dialogmodellierung, Petrietz-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
- 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

Zugeordnete Lehrveranstaltungen:

Oberseminar zu Lehrprofessur für Informatik

Forschungsthemen des Lehrstuhls: Petri Nets, Process Mining, Concurrent Systems

Prüfung

Vortrag und Abschlussbericht

Praktikum

Modul INF-0119: Projektmodul Programmiermethodik und Multimediale Informationssysteme		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Bernhard Möller		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet "Programmiermethodik und Multimediale Informationssysteme" und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; 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>Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Moduleile		
<p>Modulteil: Projektmodul Programmiermethodik und Multimediale Informationssysteme Lehrformen: Praktikum Sprache: Deutsch Angebotshäufigkeit: nach Bedarf SWS: 1</p>		
<p>Inhalte: Anwendung und Erweiterung von Kleene-Algebren, Halbringtheorie und automatisches Beweisen, Datenbanken und Informationssysteme</p>		
<p>Literatur: aktuelle Forschungspaper</p>		
<p>Zugeordnete Lehrveranstaltungen: Oberseminar für Programmiermethodik und Multimediale Informationssysteme</p>		
<p>Prüfung Projektabnahme, Vortrag und Abschlussbericht Praktikum</p>		

Modul INF-0137: Projektmodul Software- und Systems Engineering		10 ECTS/LP
Version 1.1.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität aus dem Gebiet des Software und Systems Engineering und verfügen dort über tiefergehende Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Fertigkeit zum logischen, analytischen und konzeptionellen Denken • Eigenständige Arbeit mit auch englischsprachiger Fachliteratur • 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>Arbeitsaufwand: Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
<p>Modulteil: Projektmodul Software- und Systems Engineering Lehrformen: Praktikum Sprache: Deutsch SWS: 1</p>		
Inhalte: Mitarbeit an aktuellen Forschungsthemen des Lehrstuhls		
Literatur: Abhängig von dem konkreten Projekt: Wissenschaftliche Papiere, Dokumentation		
<p>Zugeordnete Lehrveranstaltungen: Oberseminar Software- und Systems Engineering Es werden aktuelle Themen aus dem Bereich der Softwaretechnik vorgetragen und diskutiert.</p>		

Prüfung

Projektmodul Software- und Systems Engineering Projektabschluss

praktische Prüfung / Bearbeitungsfrist: 2 Monate

Modul INF-0154: Projektmodul Systemnahe Informatik und Kommunikationssysteme		10 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet der Systemnahen Informatik und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; 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>Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Projektmodul Systemnahe Informatik und Kommunikationssysteme		
Lehrformen: Praktikum		
Sprache: Deutsch		
SWS: 1		
Inhalte: Autonome Mitarbeit an aktuelle Forschungsthemen.		
Literatur: wissenschaftliche Papiere, Handbücher		
Zugeordnete Lehrveranstaltungen:		
Oberseminar Systemnahe Informatik und Kommunikationssysteme		
Es werden aktuelle Themen aus dem Bereich der Systemnahen Informatik und Kommunikationssysteme vorgetragen und diskutiert. Dazu zählen bei Lst. Knorr Selbstorganisierende Kommunikationssysteme und Next Generation Networks (NGN) und bei Lst. Ungerer Prozessorarchitekturen und Parallelisierung für harte Echtzeitsysteme, Fehlertoleranz und Transaktionsspeicher.		

Prüfung

Vortrag und schriftliche Ausarbeitung

Praktikum

Modul INF-0165: Projektmodul Theorie verteilter Systeme		10 ECTS/LP
Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Walter Vogler		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet "Theorie verteilter Systeme" und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; Verständliche, sichere und überzeugende Präsentation von Ideen, Konzepten und Ergebnissen; Qualitätsbewußtsein; Kommunikationsfähigkeit; Grundsätze guter wissenschaftlicher Praxis; Projektmanagementfähigkeiten; Wissenschaftliche Methodik</p>		
<p>Arbeitsaufwand: Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Projektmodul Theorie verteilter Systeme		
Sprache: Deutsch		
SWS: 1		
Inhalte: aktuelle Forschungsthemen in der Theorie verteilter Systeme		
Literatur: wissenschaftliche Papiere, evtl. Handbücher		
Zugeordnete Lehrveranstaltungen: Oberseminar zu Grundlagen Reaktiver Systeme		
Prüfung		
Schriftliche Ausarbeitung Projektarbeit		

Modul INF-0170: Projektmodul Human-Centered Multimedia		10 ECTS/LP
Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet "Human-Centered Multimedia" und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; 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>Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Projektmodul Human-Centered Multimedia		
Lehrformen: Praktikum		
Sprache: Deutsch		
SWS: 1		
Inhalte: Mitarbeit an aktuellen Forschungsthemen.		
Literatur: Literaturhinweise werden je nach Thema zu Beginn des Moduls gegeben.		
Zugeordnete Lehrveranstaltungen:		
Oberseminar Human-Centered Multimedia		
Prüfung		
Projektabschluss und Vortrag Praktikum		

Modul INF-0275: Projektmodul Embedded Intelligence for Health Care and Wellbeing		10 ECTS/LP
Version 1.1.0 (seit WS17/18) Modulverantwortliche/r: Prof. Dr. Björn Schuller		
<p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet der eingebetteten Systeme und der intelligenten Signalanalyse, insbesondere für Anwendungen in der Medizin- und Sportinformatik, und verfügen dort über tiefergehende Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Forschungsprojekten entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Darüber hinaus verfügen die Studierenden über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem Gebiet zu diskutieren, Zwischenziele zu definieren, sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Eigenständige Arbeit mit englischsprachiger Fachliteratur; 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; Softwareentwicklung und -test.</p>		
<p>Arbeitsaufwand: Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium)</p>		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester: ab dem 1.	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: siehe PO des Studiengangs	
Modulteile		
Modulteil: Projektmodul Embedded Intelligence for Health Care and Wellbeing		
Lehrformen: Praktikum		
Sprache: Deutsch / Englisch		
SWS: 1		
Inhalte: Autonome Mitarbeit an aktuelle Forschungsthemen.		
Literatur: Wissenschaftliche Veröffentlichungen; Handbücher; wird vom Lehrstuhl zur Verfügung gestellt.		
Zugeordnete Lehrveranstaltungen: Oberseminar Embedded Intelligence for Health Care and Wellbeing		
Prüfung		
Vortrag und schriftliche Ausarbeitung Praktikum		

Modul INF-0003: Masterarbeit		30 ECTS/LP
Version 1.0.0 (seit WS13/14) Modulverantwortliche/r: Alle Professorinnen und Professoren, die Veranstaltungen für diesen Studiengang anbieten.		
<p>Lernziele/Kompetenzen:</p> <p>Die Studierenden sind mit der wissenschaftlichen Methodik sowie Techniken der Literaturrecherche vertraut, sind in der Lage, unter Anleitung vertiefte praktische oder theoretische Methoden zur Bearbeitung eines vorgegebenen Themas einzusetzen und besitzen die Kompetenz, ein wissenschaftlich anspruchsvolles Problem der Informatik innerhalb einer vorgegebenen Frist weitgehend selbständig mit wissenschaftlichen Methoden zu bearbeiten sowie die Ergebnisse schriftlich und mündlich verständlich und überzeugend darzustellen.</p> <p>Außerdem verfügen die Studierenden über tiefergehende Fachkenntnisse und Fähigkeiten auf dem Gebiet der Masterarbeit, können Konzepte, Methoden, Verfahren, Techniken und Technologien dieses Gebiets in Forschungs- oder Anwendungsprojekten weiterentwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und ihren eigenen wissenschaftlichen Beitrag auf diesem Gebiet zu leisten. Die Studierenden haben die Fertigkeit zur Analyse und Strukturierung von Problemstellungen auf dem Gebiet der Masterarbeit und die Fertigkeit zur Entwicklung und Umsetzung von Lösungsstrategien, und kennen praxisrelevante Aufgabenstellungen. Sie kennen Vor- und Nachteile von Lösungsalternativen, können diese im jeweiligen Anwendungszusammenhang mit geeigneten Methoden wissenschaftlich aussagekräftig bewerten und geeignete Lösungsalternativen sicher auswählen. Die Studierenden haben die Fähigkeit, ihr Wissen auf dem Gebiet der Masterarbeit und verwandter Gebiete selbstständig zu erweitern und zu vertiefen.</p> <p>Darüber hinaus können die Studierenden weiterführende 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>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken, 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, Qualitätsbewusstsein, Akribie</p>		
<p>Arbeitsaufwand:</p> <p>Gesamt: 900 Std. 900 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p>		
<p>Voraussetzungen:</p> <p>keine</p>		
<p>Angebotshäufigkeit: jedes Semester</p>	<p>Empfohlenes Fachsemester:</p> <p>4.</p>	<p>Minimale Dauer des Moduls:</p> <p>1 Semester</p>
<p>SWS:</p> <p>0</p>	<p>Wiederholbarkeit:</p> <p>siehe PO des Studiengangs</p>	
<p>Modulteile</p>		
<p>Modulteil: Masterarbeit</p> <p>Sprache: Deutsch</p>		
<p>Inhalte:</p> <p>entsprechend dem gewählten Thema</p>		

Literatur:

Die Festlegung der Literatur erfolgt abhängig vom konkreten Thema der Arbeit in Absprache mit dem Betreuer.

Prüfung

Masterarbeit

Masterarbeit

Modul INF-0221: Anleitung zu wissenschaftlichen Arbeiten		ECTS/LP
Version 1.0.0 Modulverantwortliche/r:		
Lernziele/Kompetenzen: 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.		
Bemerkung: Dies ist eine freiwillige Veranstaltung und gibt keine ECTS-Punkte!		
Arbeitsaufwand: Gesamt: 15 Std.		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester:	Minimale Dauer des Moduls: 1 Semester
SWS: 1	Wiederholbarkeit: keine	

Modulteile
Modulteil: Anleitung zu wissenschaftlichen Arbeiten Sprache: Deutsch SWS: 1
Inhalte: Begleitung bei der Anfertigung von Seminar-/Bachelor-/Master-/Diplomarbeiten und Dissertationen.
Zugeordnete Lehrveranstaltungen: Anleitung zu wissenschaftlichen Arbeiten Anleitung zu wissenschaftlichen Arbeiten Anleitung zu wissenschaftlichen Arbeiten Anleitung zu wissenschaftlichen Arbeiten Software- und Systems Engineering

Modul INF-0222: Oberseminar Informatik		ECTS/LP
Version 1.0.0 Modulverantwortliche/r:		
Lernziele/Kompetenzen: 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.		
Bemerkung: Dies ist eine freiwillige Veranstaltung und gibt keine ECTS-Punkte!		
Arbeitsaufwand: Gesamt: 30 Std. 30 Std. Seminar (Präsenzstudium)		
Voraussetzungen: keine		
Angebotshäufigkeit: jedes Semester	Empfohlenes Fachsemester:	Minimale Dauer des Moduls: 1 Semester
SWS: 2	Wiederholbarkeit: keine	

Modulteile
<p>Modulteil: Oberseminar Informatik</p> <p>Lehrformen: Seminar</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p>
<p>Zugeordnete Lehrveranstaltungen:</p> <p>Oberseminar Datenbanken und Informationssysteme</p> <p>Oberseminar Embedded Intelligence for Health Care and Wellbeing</p> <p>Oberseminar Human-Centered Multimedia</p> <p>Oberseminar Multimedia Computing</p> <p>Oberseminar Organic Computing Weiterführende Themen aus dem Bereich Organic Computing</p> <p>Oberseminar Software- und Systems Engineering Es werden aktuelle Themen aus dem Bereich der Softwaretechnik vorgetragen und diskutiert.</p> <p>Oberseminar Systemnahe Informatik und Kommunikationssysteme Es werden aktuelle Themen aus dem Bereich der Systemnahen Informatik und Kommunikationssysteme vorgetragen und diskutiert. Dazu zählen bei Lst. Knorr Selbstorganisierende Kommunikationssysteme und Next Generation Networks (NGN) und bei Lst. Ungerer Prozessorarchitekturen und Parallelisierung für harte Echtzeitsysteme, Fehlertoleranz und Transaktionsspeicher.</p> <p>Oberseminar Theoretische Informatik</p> <p>Oberseminar für Programmiermethodik und Multimediale Informationssysteme</p> <p>Oberseminar zu Grundlagen Reaktiver Systeme</p> <p>Oberseminar zu Lehrprofessur für Informatik Forschungsthemen des Lehrstuhls: Petri Nets, Process Mining, Concurrent Systems</p> <p>Oberseminar zu Softwaremethodik für verteilte Systeme</p>

Oberseminar zur Mechatronik

Oberseminar zur Produktionsinformatik

Oberseminar zur Regelungstechnik