
Modulhandbuch

M.Sc. Ingenieurinformatik, PO 2016

Fakultät für Angewandte Informatik

Wintersemester 2020/2021

Studienbeginn ab WiSe 2016/2017

Wichtige Zusatzinformation für das WS 2020/21 aufgrund der Corona-Pandemie:

Bitte berücksichtigen Sie, dass aufgrund der Entwicklungen der Corona-Pandemie die Angaben zu den jeweiligen Prüfungsformaten in den Modulhandbüchern ggf. noch nicht aktuell sind. Welche Prüfungsformate schließlich bei welchen Modulen möglich sein werden, wird im weiteren Verlauf des Semesters geklärt und festgelegt werden. Entsprechende Informationen werden spätestens am 01.12.2020 bekannt gegeben.

Liebe Studierenden,

auch wenn COVID-19 uns vor besondere Herausforderungen stellt, tun wir unser bestes, um euch weiterhin ein breit aufgestelltes Lehrangebot zu bieten. Ihr findet also ähnlich viele Veranstaltungen im Modulhandbuch wie in einem "normalen" Semester auch.

Im Lehrangebot haben sich jedoch einige Änderungen ergeben:

1. Durch die Pensionierung von Prof. Knorr zum Ende des Sommersemesters 2020 fallen die Module *INF-0084: Seminar Next Generation Networks* und *INF-0085: Projektmodul Kommunikationssysteme* ersatzlos weg.
2. Außerdem sind folgende Module neu hinzu gekommen:
 - INF-0364: Seminar Software Engineering in sicherheitskritischen Systemen (MA)
 - INF-0367: Advanced Machine Learning and Computer Vision
 - INF-0368: Embedded Hardware

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 hallo@fachschaft-info.de und persönlich im Raum 1007N.

Viele Grüße,

Euer Modulhandbuch-Beauftragter

Martin Frieb

Übersicht nach Modulgruppen

1) M.Sc. Ingenieurinformatik

a) Software und Systems Engineering (ECTS: 18)

"je 18 Leistungspunkte aus den Modulgruppen Software und Systems Engineering Technische Informatik und Adaptive Systeme sowie dem Wahlbereich Software und Systems Engineering / Technische Informatik und Adaptive Systeme"

| | |
|---|----|
| INF-0037: Praktikum Automotive Software Engineering (6 ECTS/LP, Wahlpflicht)..... | 7 |
| INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme (10 ECTS/LP, Wahlpflicht) *..... | 9 |
| INF-0129: Softwaretechnik II (8 ECTS/LP, Wahlpflicht) *..... | 10 |
| INF-0131: Software- und Systemsicherheit (8 ECTS/LP, Wahlpflicht)..... | 13 |
| INF-0136: Seminar Software- und Systems Engineering (Master) (4 ECTS/LP, Wahlpflicht) *..... | 15 |
| INF-0137: Projektmodul Software- und Systems Engineering (10 ECTS/LP, Wahlpflicht) *..... | 17 |
| INF-0189: Qualitätssicherung im Software Engineering (5 ECTS/LP, Wahlpflicht)..... | 19 |
| INF-0217: Praktikum Autonomes Fahren (10 ECTS/LP, Wahlpflicht)..... | 21 |
| INF-0232: Seminar Medical Information Sciences (MA) (4 ECTS/LP, Wahlpflicht) *..... | 23 |
| INF-0233: Industrierobotik (8 ECTS/LP, Wahlpflicht) *..... | 25 |
| INF-0234: Flugrobotik (5 ECTS/LP, Wahlpflicht)..... | 27 |
| INF-0235: Software für Industrie 4.0 (5 ECTS/LP, Wahlpflicht)..... | 29 |
| INF-0248: Kollaborative Robotik (8 ECTS/LP, Wahlpflicht) *..... | 31 |
| INF-0307: Modellbasierte Entwicklung und Analyse von Software Systemen (6 ECTS/LP, Wahlpflicht)..... | 34 |
| INF-0344: Seminar Software Engineering verteilter Systeme (MA) (4 ECTS/LP, Wahlpflicht) *..... | 36 |
| INF-0346: Seminar Automotive Software and Systems Engineering (MA) (4 ECTS/LP, Wahlpflicht) *..... | 38 |
| INF-0348: Seminar Avionic Software and Systems Engineering (MA) (4 ECTS/LP, Wahlpflicht)..... | 40 |
| INF-0364: Seminar Software Engineering in sicherheitskritischen Systemen (MA) (4 ECTS/LP, Wahlpflicht) *..... | 42 |

b) Technische Informatik und Adaptive Systeme (ECTS: 18)

"je 18 Leistungspunkte aus den Modulgruppen Software und Systems Engineering Technische Informatik und Adaptive Systeme sowie dem Wahlbereich Software und Systems Engineering / Technische Informatik und Adaptive Systeme"

| | |
|--|----|
| INF-0066: Organic Computing II (5 ECTS/LP, Wahlpflicht)..... | 44 |
|--|----|

| | |
|--|----|
| INF-0067: Peer-to-Peer und Cloud Computing (5 ECTS/LP, Wahlpflicht)..... | 46 |
| INF-0069: Weiterführende Betriebssystemkonzepte (8 ECTS/LP, Wahlpflicht)..... | 48 |
| INF-0070: Seminar Organic Computing (4 ECTS/LP, Wahlpflicht) *..... | 50 |
| INF-0071: Seminar Naturanaloge Algorithmen und Multiagentensysteme (4 ECTS/LP, Wahlpflicht)..... | 52 |
| INF-0072: Projektmodul Organic Computing (10 ECTS/LP, Wahlpflicht) *..... | 54 |
| INF-0088: Bayesian Networks (5 ECTS/LP, Wahlpflicht)..... | 56 |
| INF-0092: Multimedia II: Machine Learning and Computer Vision (8 ECTS/LP, Wahlpflicht)..... | 58 |
| INF-0093: Probabilistic Robotics (5 ECTS/LP, Wahlpflicht)..... | 60 |
| INF-0133: Selbstorganisierende, adaptive Systeme (8 ECTS/LP, Wahlpflicht) *..... | 62 |
| INF-0147: Prozessorarchitektur (5 ECTS/LP, Wahlpflicht)..... | 64 |
| INF-0149: Praktikum Eingebettete Systeme (5 ECTS/LP, Wahlpflicht)..... | 66 |
| INF-0150: Hardware-Entwurf (8 ECTS/LP, Wahlpflicht)..... | 68 |
| INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung (8 ECTS/LP, Wahlpflicht)..... | 70 |
| INF-0216: Vertiefte Multicore-Programmierung (8 ECTS/LP, Wahlpflicht)..... | 71 |
| INF-0251: Seminar Artificial Intelligence (4 ECTS/LP, Wahlpflicht)..... | 73 |
| INF-0291: Praktikum Selbstlernende Systeme (8 ECTS/LP, Wahlpflicht) *..... | 75 |
| INF-0297: Praktikum Prozessorbau (5 ECTS/LP, Wahlpflicht)..... | 77 |
| INF-0309: Echtzeitsysteme (8 ECTS/LP, Wahlpflicht) *..... | 79 |
| INF-0335: Safety-Critical Systems (5 ECTS/LP, Wahlpflicht)..... | 81 |
| INF-0337: Seminar Embedded Systems (Master) (4 ECTS/LP, Wahlpflicht) *..... | 83 |
| INF-0340: Projektmodul Embedded Systems (10 ECTS/LP, Wahlpflicht) *..... | 85 |
| INF-0367: Advanced Machine Learning and Computer Vision (5 ECTS/LP, Wahlpflicht) *..... | 87 |
| INF-0368: Embedded Hardware (5 ECTS/LP, Wahlpflicht) *..... | 89 |

c) Wahlbereich Software und Systems Engineering / Technische Informatik und Adaptive Systeme (ECTS: 18)

"je 18 Leistungspunkte aus den Modulgruppen Software und Systems Engineering Technische Informatik und Adaptive Systeme sowie dem Wahlbereich Software und Systems Engineering / Technische Informatik und Adaptive Systeme"

| | |
|--|----|
| INF-0037: Praktikum Automotive Software Engineering (6 ECTS/LP, Wahlpflicht)..... | 91 |
| INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme (10 ECTS/LP, Wahlpflicht) *..... | 93 |
| INF-0066: Organic Computing II (5 ECTS/LP, Wahlpflicht)..... | 94 |

| | |
|--|-----|
| INF-0067: Peer-to-Peer und Cloud Computing (5 ECTS/LP, Wahlpflicht)..... | 96 |
| INF-0069: Weiterführende Betriebssystemkonzepte (8 ECTS/LP, Wahlpflicht)..... | 98 |
| INF-0070: Seminar Organic Computing (4 ECTS/LP, Wahlpflicht) *..... | 100 |
| INF-0071: Seminar Naturanaloge Algorithmen und Multiagentensysteme (4 ECTS/LP, Wahlpflicht)..... | 102 |
| INF-0072: Projektmodul Organic Computing (10 ECTS/LP, Wahlpflicht) *..... | 104 |
| INF-0088: Bayesian Networks (5 ECTS/LP, Wahlpflicht)..... | 106 |
| INF-0092: Multimedia II: Machine Learning and Computer Vision (8 ECTS/LP, Wahlpflicht)..... | 108 |
| INF-0093: Probabilistic Robotics (5 ECTS/LP, Wahlpflicht)..... | 110 |
| INF-0129: Softwaretechnik II (8 ECTS/LP, Wahlpflicht) *..... | 112 |
| INF-0131: Software- und Systemsicherheit (8 ECTS/LP, Wahlpflicht)..... | 115 |
| INF-0133: Selbstorganisierende, adaptive Systeme (8 ECTS/LP, Wahlpflicht) *..... | 117 |
| INF-0136: Seminar Software- und Systems Engineering (Master) (4 ECTS/LP, Wahlpflicht) *..... | 119 |
| INF-0137: Projektmodul Software- und Systems Engineering (10 ECTS/LP, Wahlpflicht) *..... | 121 |
| INF-0147: Prozessorarchitektur (5 ECTS/LP, Wahlpflicht)..... | 123 |
| INF-0149: Praktikum Eingebettete Systeme (5 ECTS/LP, Wahlpflicht)..... | 125 |
| INF-0150: Hardware-Entwurf (8 ECTS/LP, Wahlpflicht)..... | 127 |
| INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung (8 ECTS/LP, Wahlpflicht)..... | 129 |
| INF-0189: Qualitätssicherung im Software Engineering (5 ECTS/LP, Wahlpflicht)..... | 130 |
| INF-0216: Vertiefte Multicore-Programmierung (8 ECTS/LP, Wahlpflicht)..... | 132 |
| INF-0217: Praktikum Autonomes Fahren (10 ECTS/LP, Wahlpflicht)..... | 134 |
| INF-0232: Seminar Medical Information Sciences (MA) (4 ECTS/LP, Wahlpflicht) *..... | 136 |
| INF-0233: Industrierobotik (8 ECTS/LP, Wahlpflicht) *..... | 138 |
| INF-0234: Flugrobotik (5 ECTS/LP, Wahlpflicht)..... | 140 |
| INF-0235: Software für Industrie 4.0 (5 ECTS/LP, Wahlpflicht)..... | 142 |
| INF-0248: Kollaborative Robotik (8 ECTS/LP, Wahlpflicht) *..... | 144 |
| INF-0251: Seminar Artificial Intelligence (4 ECTS/LP, Wahlpflicht)..... | 147 |
| INF-0291: Praktikum Selbstlernende Systeme (8 ECTS/LP, Wahlpflicht) *..... | 149 |
| INF-0297: Praktikum Prozessorbau (5 ECTS/LP, Wahlpflicht)..... | 151 |
| INF-0307: Modellbasierte Entwicklung und Analyse von Software Systemen (6 ECTS/LP, Wahlpflicht)..... | 153 |
| INF-0309: Echtzeitsysteme (8 ECTS/LP, Wahlpflicht) *..... | 155 |

| | |
|--|-----|
| INF-0335: Safety-Critical Systems (5 ECTS/LP, Wahlpflicht)..... | 157 |
| INF-0337: Seminar Embedded Systems (Master) (4 ECTS/LP, Wahlpflicht) *..... | 159 |
| INF-0340: Projektmodul Embedded Systems (10 ECTS/LP, Wahlpflicht) *..... | 161 |
| INF-0344: Seminar Software Engineering verteilter Systeme (MA) (4 ECTS/LP, Wahlpflicht) *..... | 163 |
| INF-0346: Seminar Automotive Software and Systems Engineering (MA) (4 ECTS/LP, Wahlpflicht) *..... | 165 |
| INF-0348: Seminar Avionic Software and Systems Engineering (MA) (4 ECTS/LP, Wahlpflicht)..... | 167 |
| INF-0364: Seminar Software Engineering in sicherheitskritischen Systemen (MA) (4 ECTS/LP, Wahlpflicht) *..... | 169 |
| INF-0367: Advanced Machine Learning and Computer Vision (5 ECTS/LP, Wahlpflicht) *..... | 171 |
| INF-0368: Embedded Hardware (5 ECTS/LP, Wahlpflicht) *..... | 173 |

d) Mechatronik und Produktion (ECTS: 36)

"36 Leistungspunkte aus der Modulgruppe Mechatronik und Produktion"

| | |
|--|-----|
| INF-0236: Digitale Regelsysteme (6 ECTS/LP, Wahlpflicht) *..... | 175 |
| INF-0237: Projektmodul Regelungstechnik (10 ECTS/LP, Wahlpflicht) *..... | 178 |
| INF-0238: Digitale Fabrik (6 ECTS/LP, Wahlpflicht) *..... | 180 |
| INF-0244: Projektmodul Produktionsinformatik (10 ECTS/LP, Wahlpflicht) *..... | 182 |
| INF-0245: Intelligent vernetzte Produktion (5 ECTS/LP, Wahlpflicht)..... | 184 |
| INF-0246: Seminar Industrie 4.0 (4 ECTS/LP, Wahlpflicht) *..... | 186 |
| INF-0247: Praktikum für Produktionsinformatik (Vertiefung) (6 ECTS/LP, Wahlpflicht) *..... | 188 |
| INF-0249: Seminar zu speziellen Kapiteln der Regelungstechnik (4 ECTS/LP, Wahlpflicht) *..... | 190 |
| INF-0264: Nichtlineare Regelsysteme (5 ECTS/LP, Wahlpflicht) *..... | 192 |
| INF-0272: Intelligente Signalanalyse in der Medizin (5 ECTS/LP, Wahlpflicht)..... | 194 |
| INF-0273: Praktikum Mobile Sensing for Fitness and Wellbeing (5 ECTS/LP, Wahlpflicht)..... | 196 |
| INF-0301: Objektorientierte Methoden der Modellbildung und Simulation (6 ECTS/LP, Wahlpflicht) *..... | 198 |
| INF-0302: Projektmodul Mechatronik (10 ECTS/LP, Wahlpflicht) *..... | 200 |
| INF-0304: Seminar zu aktuellen Forschungsthemen der Mechatronik (4 ECTS/LP, Wahlpflicht) *.... | 202 |
| INF-0318: Praktikum Simulation von autonomen Fahrzeugen (Vertiefung) (8 ECTS/LP, Wahlpflicht) *..... | 204 |
| INF-0319: Praktikum Interdisziplinäres Projekt Ingenieurinformatik (6 ECTS/LP, Wahlpflicht)..... | 206 |
| INF-0333: Konzepte autonomer mechatronischer Systeme (5 ECTS/LP, Wahlpflicht)..... | 207 |

e) Abschlussarbeit (ECTS: 30)

"30 Leistungspunkte im Rahmen der Masterarbeit"

INF-0003: Masterarbeit (30 ECTS/LP, Pflicht)..... 209

2) 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) * 211

INF-0222: Oberseminar Informatik (0 ECTS/LP, Wahlfach) * 212

| | | |
|--|---|---|
| Modul INF-0037: Praktikum Automotive Software Engineering <i>Practical Module Automotive Software Engineering</i> | | 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 Fahrerassistenzsystemen mit aktuellen Methoden und Tools der modellbasierten Entwicklung eingebetteter Systeme. Die Studierenden erlangen tiefere 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> | | |
| <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: 6</p> | <p>Wiederholbarkeit: siehe PO des Studiengangs</p> | |
| Moduleile | | |
| <p>Moduleil: Praktikum Automotive Software Engineering (MA) Lehrformen: Praktikum Sprache: Deutsch Angebotshäufigkeit: jedes Wintersemester 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

Prüfung

Praktikum Automotive Software Engineering (MA)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

| | | |
|---|---|---|
| Modul INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme <i>Project Module Software Methodologies for Distributed Systems</i> | | 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 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. 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 | | |
| <p>Modulteil: Projektmodul Softwaremethodiken für verteilte Systeme Lehrformen: Praktikum Sprache: Deutsch SWS: 1</p> | | |
| Inhalte: Aktuelle Forschungsthemen an der Professur Softwaremethodik für verteilte Systeme | | |
| Literatur: Wird zu den jeweiligen Themen bereitgestellt. | | |
| <p>Zugeordnete Lehrveranstaltungen: Oberseminar zu Softwaremethodik für verteilte Systeme <i>*Veranstaltung wird online/digital abgehalten.*</i></p> | | |
| <p>Prüfung Projektabnahme, Vortrag, Abschlussbericht Praktikum</p> | | |

| | | |
|---|---|---|
| Modul INF-0129: Softwaretechnik II <i>Software Engineering II</i> | | 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 |
| <p>Modulteil: Softwaretechnik II (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p> |
| <p>Inhalte:</p> <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 |
| <p>Literatur:</p> <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 |
| <p>Zugeordnete Lehrveranstaltungen:</p> <p>Softwaretechnik 2 (Vorlesung)</p> <p><i>*Veranstaltung wird als Hybrid/gemischt abgehalten.*</i></p> <p>Die Veranstaltung findet im Wesentlichen digital statt. Vorlesungsinhalte werden als Screencast zum Download angeboten. Zusätzlich gibt es pro Woche höchstens einen synchronen Termin, bei dem Fragen zum Stoff gestellt werden können und die Übungen behandelt werden. Für den synchronen Termin ist im Stundenplan Dienstag ab 16 Uhr vorgesehen. Falls es die Corona-Situation und die Teilnehmerzahl erlauben, gibt es am Anfang des neuen Jahres noch zwei Präsenztermine, in denen Stoff praktisch erarbeitet wird.</p> |

Modulteil: Softwaretechnik II (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Softwaretechnik 2 (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Softwaretechnik II Klausur

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0131: Software- und Systemsicherheit <i>Software and Systems Security</i> | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| Lernziele/Kompetenzen: Fachbezogene Kompetenzen: 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. Sie können Bedrohungsanalyse durchführen, kryptographische Protokolle entwickeln, Chipkarten programmieren, sicherheitskritische Systeme in Teams entwerfen, und ihre Ergebnisse dokumentieren. Sie können systematisch Bedrohungen für Softwaresysteme analysieren und deren Risiken bewerten. Sie können einen modellgetriebenen Entwicklungsprozess für sicherheitskritische Systeme anwenden. Sie sind in der Lage, bedeutende technische Entwicklungen zu erkennen und fachliche Lösungskonzepte in Programme umzusetzen. Sie haben Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen. Schlüsselqualifikationen: <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 | | |
| 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: 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 | | |
| Modulteil: Software- und Systemsicherheit (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2 | | |

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-0136: Seminar Software- und Systems Engineering (Master) <i>Seminar Software- und Systems Engineering (Master)</i> | | 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. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 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 | |

| |
|---|
| Moduleile |
| <p>Moduleil: Seminar Software- und Systems Engineering (Master)</p> <p>Lehrformen: Seminar</p> <p>Sprache: Deutsch / Englisch</p> <p>Angebotshäufigkeit: jedes Sommersemester</p> <p>SWS: 2</p> |
| <p>Inhalte:</p> <p>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:</p> <p>Abhängig von den konkreten Themen des Seminars</p> |
| <p>Zugeordnete Lehrveranstaltungen:</p> <p>Seminar zu IT-Sicherheit (Seminar)</p> <p><i>*Veranstaltung wird online/digital abgehalten.*</i></p> <p>Es werden maximal die 10 unten aufgeführten Themen nach dem Prinzip First Come, First Served vergeben. Um am Seminar teilzunehmen, schicken Sie eine Mail mit ZWEI Themenwünschen an philip.lenzen@informatik.uni-augsburg.de. Folgende Seminarthemen werden angeboten: Themenblock 1: Exploits 1. Angriffe auf Signaturen von PDF-Dokumenten Das "Portable Document Format" (PDF) ist de-facto Standard für elektronische Dokumente, unter anderem für elektronische Verträge und Rechnungen im Staats- und Finanzwesen. Zur Sicherung der Authentizität und Integrität von PDF-Dokumenten werden Digitale Signaturen eingesetzt. Allerdings ist eine Vielzahl an PDF-Viewern verwundbar hinsichtlich neuartiger Angriffe auf Signaturen, die Forscher der Ruhr-Universität Bochum entdeckt haben. Diese sollen im Rahmen dieser Arbeit näher untersucht werden. 2. Automatische Heap-basierte Exploit-Generierung Typische Ansätze der automatisierten Generierung von Exploits (Befehlsketten, die aufgrund von Speicherkorruption be ... (weiter siehe Digicampus)</p> |
| <p>Prüfung</p> <p>Seminar Software- und Systems Engineering (Master)</p> <p>Seminar / Prüfungsdauer: 45 Minuten</p> <p>Bearbeitungsfrist: 3 Monate</p> |

| | | |
|---|---|---|
| Modul INF-0137: Projektmodul Software- und Systems Engineering <i>Project Module Software- and Systems Engineering</i> | | 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 | | |
| Modulteil: Projektmodul Software- und Systems Engineering | | |
| Lehrformen: Praktikum | | |
| Sprache: Deutsch / Englisch | | |
| SWS: 1 | | |
| Inhalte: Mitarbeit an aktuellen Forschungsthemen des Lehrstuhls | | |
| Literatur: Abhängig von dem konkreten Projekt: Wissenschaftliche Papiere, Dokumentation | | |
| Zugeordnete Lehrveranstaltungen: Oberseminar Software- und Systems Engineering | | |

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen am Lehrstuhl für Softwaretechnik zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren Sie direkt einen Mitarbeiter, dessen Forschung Sie interessiert: <https://www.uni-augsburg.de/de/fakultaet/fai/isse/prof/swtse/teaching/students/> Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Prüfung

Projektmodul Software- und Systems Engineering Projektabnahme

praktische Prüfung / Bearbeitungsfrist: 2 Monate

| | | |
|--|---|---|
| Modul INF-0189: Qualitätssicherung im Software Engineering <i>Quality Assurance in Software Engineering</i> | | 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 | |
| Moduleile | | |
| <p>Moduleil: 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-0217: Praktikum Autonomes Fahren <i>Practical Module Autonomous Driving</i> | | 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> | | |
| <p>Angebotshäufigkeit: jedes Sommersemester</p> | <p>Empfohlenes Fachsemester: ab dem 1.</p> | <p>Minimale Dauer des Moduls: 1 Semester</p> |
| <p>SWS: 10</p> | <p>Wiederholbarkeit: siehe PO des Studiengangs</p> | |
| <p>Moduleile</p> <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-0232: Seminar Medical Information Sciences (MA) <i>Seminar Medical Information Sciences (MA)</i> | | 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> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Medical Information Sciences (Seminar) | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| 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. | | |
| Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt. | | |
| Zugeordnete Lehrveranstaltungen: | | |

Seminar Medical Information Sciences (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

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

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

| | | |
|---|--|--|
| Modul INF-0233: Industrierobotik <i>Industrial Robotics</i> | | 8 ECTS/LP |
| Version 2.2.0 (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 Wintersemester | 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)

Veranstaltung wird als Hybrid/gemischt abgehalten.

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)

Veranstaltung wird als Hybrid/gemischt abgehalten.

Prüfung

Industrierobotik

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 3 Monate

Beschreibung:

Ausnahmefall WiSe 2020/2021: Die Prüfung findet als Portfolioprüfung statt.

| | | |
|---|---|---|
| Modul INF-0234: Flugrobotik <i>Aerial Robotics</i> | | 5 ECTS/LP |
| Version 1.3.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| <p>Lernziele/Kompetenzen: Die Studierenden kennen die Grundlagen fliegender Roboter, sind in der Lage, diese zu programmieren, und können die Eignung verschiedener Vorgehensweisen dafür bewerten. Sie verstehen den Aufbau von Flugrobotern sowie die Funktionsweise oft verwendeter Sensorik. Sie können fachliche Lösungskonzepte in Software umsetzen und dabei Entwurfsalternativen bewerten und anwenden. Die Studierenden sind in der Lage für praxisrelevante Aufgabenstellungen die Eignung fliegender Roboter zu bewerten. Sie haben die Fähigkeit zum analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikation:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten | | |
| <p>Bemerkung: "Grundlagen der Robotik" als Vorkenntnis empfohlen</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: 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: Flugrobotik (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p> | | |

Inhalte:

Fliegende Roboter, wie zum Beispiel Multicopter, sind immer häufiger im Alltag anzutreffen. Durch Fortschritte im Bereich von Mikrocontrollern, leistungsfähigen Elektromotoren und der Akku-Technologie sind fliegende Roboter für viele Anwendungsbereiche interessant geworden und werden in Zukunft noch viel stärker an Bedeutung gewinnen.

In dieser Veranstaltung liegt der Schwerpunkt auf der Entwicklung von Software für fliegende Robotersysteme. Dazu werden Grundlagen in folgenden Bereichen vermittelt:

- Aufbau und Funktionsweise von Multicoptern
- Aufbau von Flugroboter-Steuerungen
- integrierte Sensorik und deren Funktionsweise
- Verarbeitung verrauschter Sensordaten
- Grundlagen räumlicher Beschreibungsformalismen
- Steuerungsmodi von Flugrobotern und autonome Grundfähigkeiten
- Rechtliche Grundlagen für den Einsatz von Flugrobotern

Auf Basis dieser Grundlagen werden verschiedene praktische Aufgabenstellungen prototypisch umgesetzt. Die Softwareentwicklung erfolgt in objektorientierten Programmiersprachen. Die Prototypen werden im Rahmen der begleitenden Übung evaluiert und optimiert.

Literatur:

- Probabilistic Robotics. Sebastian Thrun, Wolfram Burgard and Dieter Fox. MIT Press, 2005.
- Planning Algorithms. Steven M. LaValle. Cambridge University Press, 2006.

Modulteil: Flugrobotik (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Flugrobotik

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 3 Monate

| | | |
|---|---|---|
| Modul INF-0235: Software für Industrie 4.0 <i>Software for Industry 4.0</i> | | 5 ECTS/LP |
| Version 1.2.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| <p>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.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenzen • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten | | |
| <p>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)</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: Software für Industrie 4.0 (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p> | | |
| <p>Inhalte: Die Vorlesung behandelt aktuelle Themen im Kontext von Industrie 4.0. Dazu zählen folgende Inhalte:</p> <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 | | |
| <p>Literatur:</p> <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 <i>Collaborative Robotics</i> | | 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> | | |
| Voraussetzungen: keine | | ECTS/LP-Bedingungen: Bestehen der Modulprüfung |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Moduleile | | |
| <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)**

Veranstaltung wird als Hybrid/gemischt abgehalten.

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)**

Veranstaltung wird als Hybrid/gemischt abgehalten.

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-0307: Modellbasierte Entwicklung und Analyse von Software Systemen <i>Model-Based Development and Analysis of Software Systems</i> | | 6 ECTS/LP |
| Version 1.1.0 (seit SoSe19) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer | | |
| <p>Lernziele/Kompetenzen: Modellbasierte Entwicklung und Analyse von Software Systemen 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. 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> | | |
| Voraussetzungen: Frühere Veranstaltung "Modellgetriebene Softwareentwicklung" darf nicht belegt worden sein wegen Überschneidungen. | | |
| Angebotshäufigkeit: jedes Sommersemester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 5 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Modellbasierte Entwicklung und Analyse von Software Systemen (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 3 | | |
| Inhalte: Modellbasierte Entwicklung und Analyse von Software Systemen befasst sich mit der Effizienzsteigerung in der Softwareherstellung durch Automatisierung und Wiederverwendung. Dabei werden Infrastrukturcode, Subsysteme, Konfigurationen oder ganze Anwendungen aus Modellen generiert. | | |

Literatur:

- 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: Modellbasierte Entwicklung und Analyse von Software Systemen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Modellbasierte Entwicklung und Analyse von Software Systemen

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

| | | |
|---|---|---|
| Modul INF-0344: Seminar Software Engineering verteilter Systeme (MA) <i>Seminar Software Engineering of Distributed Systems (MA)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe20) 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. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p> | | |
| Voraussetzungen: Frühere Veranstaltung "Seminar über Software Engineering verteilter Systeme (MA)" (INF-0039) darf nicht belegt worden sein wegen Überschneidungen. | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Software Engineering verteilter Systeme (MA) | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| Inhalte: Aktuelle Software Engineering-Themen aus Industrie und Forschung. | | |
| Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt. | | |

Zugeordnete Lehrveranstaltungen:

Seminar Software Engineering verteilter Systeme (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

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

Seminar Software Engineering verteilter Systeme (MA)

Schriftlich-Mündliche Prüfung

| | | |
|--|---|---|
| Modul INF-0346: Seminar Automotive Software and Systems Engineering (MA) <i>Seminar Automotive Software and Systems Engineering (MA)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer | | |
| 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 & Systems 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. | | |
| 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 | | |
| Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium) | | |
| Voraussetzungen: Frühere Veranstaltung "Seminar Grundlagen des Software Engineering für Automotive Systems (MA)" (INF-0040) darf nicht belegt worden sein wegen Überschneidungen. | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Automotive Software and Systems Engineering (MA) Lehrformen: Seminar Sprache: Deutsch SWS: 2 | | |
| Inhalte: Aktuelle Software Engineering-Themen aus Industrie und Forschung. | | |

Literatur:

Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.

Zugeordnete Lehrveranstaltungen:

Seminar Automotive Software and Systems Engineering (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

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

Seminar Automotive Software and Systems Engineering (MA)

Schriftlich-Mündliche Prüfung

| | | |
|---|---|---|
| Modul INF-0348: Seminar Avionic Software and Systems Engineering (MA) <i>Seminar Avionic Software and Systems Engineering (MA)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe20) 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 & Systems 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: Frühere Veranstaltung "Seminar Grundlagen des Software Engineering für Avionic Systems (MA)" (INF-0041) darf nicht belegt worden sein wegen Überschneidungen.</p> | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Avionic Software and Systems Engineering (MA) | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| Inhalte: Aktuelle Software Engineering-Themen aus Industrie und Forschung. | | |

Literatur:

Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.

Prüfung

Seminar Avionic Software and Systems Engineering (MA)

Schriftlich-Mündliche Prüfung

| | | |
|---|---|---|
| Modul INF-0364: Seminar Software Engineering in sicherheitskritischen Systemen (MA) <i>Seminar Software Engineering in Safety- and Security-Critical Systems (MA)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit WS20/21) 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 in sicherheitskritische Systemen und deren verwandten Disziplinen 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: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Software Engineering in sicherheitskritischen Systemen (MA) | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| Inhalte: Aktuelle Software Engineering-Themen aus Industrie und Forschung. | | |
| Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt. | | |
| Zugeordnete Lehrveranstaltungen: | | |

Seminar Software Engineering in sicherheitskritischen Systemen (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

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

Seminar Software Engineering in sicherheitskritischen Systemen (MA)

Schriftlich-Mündliche Prüfung

| | | |
|---|---|---|
| Modul INF-0066: Organic Computing II <i>Organic Computing II</i> | | 5 ECTS/LP |
| Version 1.3.0 (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 |
| <p>Modulteil: Organic Computing II (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p> |
| <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 <i>Peer-to-Peer and Cloud Computing</i> | | 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 | |
| Moduleile | | |
| <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

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

Lehrformen: Übung

Sprache: Deutsch

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

SWS: 2

Prüfung

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

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

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

| | | |
|---|---|---|
| Modul INF-0069: Weiterführende Betriebssystemkonzepte <i>Advanced Operating System Concepts</i> | | 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-0070: Seminar Organic Computing <i>Seminar Organic Computing</i> | | 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 | |
| Modulteile | | |
| 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)

Veranstaltung wird online/digital abgehalten.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

| | | |
|--|---|---|
| Modul INF-0071: Seminar Naturalogische Algorithmen und Multiagentensysteme <i>Seminar Nature inspired Algorithms and Multi Agent Systems</i> | | 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-0072: Projektmodul Organic Computing <i>Project Module Organic Computing</i> | | 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 | | |
| <p>Modulteil: Projektmodul Organic Computing Lehrformen: Praktikum Sprache: Deutsch SWS: 1</p> | | |
| Inhalte: Mitarbeit an aktuellen Forschungsthemen. | | |
| <p>Literatur: In Abhängigkeit vom zu bearbeitenden Thema:</p> <ul style="list-style-type: none"> • Paper • Buch • Handbuch | | |
| <p>Zugeordnete Lehrveranstaltungen: Oberseminar Organic Computing <i>*Veranstaltung wird online/digital abgehalten.*</i> Weiterführende Themen aus dem Bereich Organic Computing</p> | | |

Prüfung

Vortrag und Abschlussbericht

Praktikum

| | | |
|--|---|---|
| Modul INF-0088: Bayesian Networks <i>Bayesian Networks</i> | | 5 ECTS/LP |
| Version 1.0.0 (seit WS15/16) 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> | | |
| <p>Angebotshäufigkeit: jedes Sommersemester</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: 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 <i>Multimedia II: Machine Learning and Computer Vision</i> | | 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 zum 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 | | |
| Modulteil: Multimedia II: Machine Learning and Computer Vision (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 4 | | |
| <p>Inhalte: Die Vorlesung gibt einen guten Überblick über alle Aspekte des maschinellen Lernens und der maschinellen Extraktion von Informationen aus Multimediadaten (z.B. "Google Image Search", "Google Goggles"). Die erlernten Konzepte werden in den Übungen anhand von erfolgreichen Beispielen aus der Praxis ausprobiert, geübt, analysiert und bewertet. Zum Ende des Semesters werden fortgeschrittene Themen wie Objektdetektion und Objekterkennung von Gesichtern und Menschen behandelt. Die Inhalte der Vorlesung umfassen: Machine Learning (Decision Tree Learning, Artificial Neural Networks, Bayesian Learning, Discrete Adaboost), Data Reduction (Quantization (K-Means Clustering, Affinity Propagation), Dimensionality Reduction Techniques (PCA, NMF, Random Projection, MDS)) und Image Processing & Computer Vision (Salient Feature Points and Feature Descriptors, Object Detection (Face/Car/People Detection), Object Recognition (Face Recognition), Image Search with pLSA)</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 <i>Probabilistic Robotics</i> | | 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: 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: Probabilistic Robotics (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch Angebotshäufigkeit: jedes Wintersemester SWS: 2</p> | | |

Inhalte:

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.

Modulteil: Probabilistic Robotics (Übung)

Lehrformen: Übung

Sprache: Deutsch

Angebotshäufigkeit: jedes Wintersemester

SWS: 2

Prüfung

Probabilistic Robotics (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

| | | |
|---|---|---|
| Modul INF-0133: Selbstorganisierende, adaptive Systeme <i>Self-organizing, adaptive systems</i> | | 8 ECTS/LP |
| Version 3.0.0 (seit WS20/21) 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)

Veranstaltung wird online/digital abgehalten.

"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" sowie flexible Produktion (<https://www.uni-augsburg.de/de/fakultaet/fai/isse/projects/soproduction/>) und Planung und Selbstorganisation in mobilen Multi-Roboter-Systemen (<https://www.uni-augsburg.de/de/fakultaet/fai/isse/projects/combo/>) aus.

Modulteil: Selbstorganisierende, adaptive Systeme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Selbst-organisierende, adaptive Systeme (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Selbstorganisierende, adaptive Systeme

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Beschreibung:

Ausnahmefall WiSe 2020/2021: Die Prüfung findet als Klausur statt.

| | | |
|--|---|---|
| Modul INF-0147: Prozessorarchitektur <i>Processor Architecture</i> | | 5 ECTS/LP |
| Version 2.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer Dr. Martin Frieb | | |
| <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. 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Übung (Präsenzstudium)</p> | | |
| Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen | | |
| Angebotshäufigkeit: unregelmäßig | 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 superskalärer 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

Modulteil: Prozessorarchitektur (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Prozessorarchitektur

Klausur / Prüfungsdauer: 60 Minuten

| | | |
|--|--|--|
| Modul INF-0149: Praktikum Eingebettete Systeme <i>Practical Module Embedded Systems</i> | | 5 ECTS/LP |
| Version 2.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <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. 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Praktikum (Präsenzstudium)</p> | | |
| <p>Voraussetzungen: Kenntnisse in C.</p> | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| <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

Praktikum Eingebettete Systeme

praktische Prüfung

| | | |
|---|---|---|
| Modul INF-0150: Hardware-Entwurf <i>Hardware Design</i> | | 8 ECTS/LP |
| Version 2.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer Dr. Martin Frieb | | |
| <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>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: Modul Systemnahe Informatik (INF-0138) - empfohlen | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Hardware-Entwurf (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| <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

Hardware-Entwurf

praktische Prüfung

Beschreibung:

Erfolgreiche Praktikumsteilnahme, Projektvorstellung am Ende des Semesters

| | | |
|---|---|---|
| Modul INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung <i>Practical Module Multimodal Real Time Signal Processing</i> | | 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> | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| <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 Praktikum Multimodale Echtzeitsignalverarbeitung praktische Prüfung</p> | | |

| | | |
|--|---|---|
| Modul INF-0216: Vertiefte Multicore-Programmierung <i>Advanced Multicore Programming</i> | | 8 ECTS/LP |
| Version 2.0.0 (seit SoSe20 bis WS20/21) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <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 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. 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)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Vertiefte Multicore-Programmierung (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| 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. | | |

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

Portfolioprufung

Beschreibung:

Projektvorstellung und -abnahme, Fragen zu Vorlesung und ubung

| | | |
|--|---|--|
| Modul INF-0251: Seminar Artificial Intelligence <i>Seminar Artificial Intelligence</i> | | 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 | |
| Moduleile | | |
| Modulteil: Seminar Artificial Intelligence Lehrformen: Seminar Sprache: Deutsch / Englisch SWS: 2 ECTS/LP: 4.0 | | |

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

Prüfung

Seminar Artificial Intelligence

Schriftlich-Mündliche Prüfung

| | | |
|---|---|--|
| Modul INF-0291: Praktikum Selbstlernende Systeme <i>Practical Module Self-Learning Systems</i> | | 8 ECTS/LP |
| Version 1.2.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| 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. | | |
| Schlüsselqualifikationen: <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 | | |
| Arbeitsaufwand: Gesamt: 240 Std. 225 Std. Praktikum (Selbststudium) 15 Std. Teilnahme an Lehrveranstaltungen (Präsenzstudium) | | |
| Voraussetzungen: Empfohlen: die erfolgreiche Teilnahme an der Vorlesung OC2, Programmiererfahrung, Teamfähigkeit Modul Organic Computing II (INF-0066) - empfohlen | | ECTS/LP-Bedingungen: Bestehen der Modulprüfung |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: ab dem 2. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Praktikum Selbstlernende Systeme Lehrformen: Praktikum Dozenten: Prof. Dr. Jörg Hähner Sprache: Deutsch SWS: 2 ECTS/LP: 8.0 | | |

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)

Veranstaltung wird online/digital abgehalten.

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-0297: Praktikum Prozessorbau <i>Practical Module Processor Design</i> | | 5 ECTS/LP |
| Version 2.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer Dr. Martin Frieb | | |
| <p>Lernziele/Kompetenzen:</p> <p>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>Arbeitsaufwand:</p> <p>Gesamt: 150 Std.</p> <p>90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>60 Std. Praktikum (Präsenzstudium)</p> | | |
| <p>Voraussetzungen:</p> <p>Modul Systemnahe Informatik (INF-0138) - empfohlen</p> <p>Modul Prozessorarchitektur (INF-0147) - empfohlen</p> | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Moduleile | | |
| <p>Modulteil: Praktikum Prozessorbau</p> <p>Lehrformen: Praktikum</p> <p>Sprache: Deutsch</p> <p>SWS: 4</p> | | |
| <p>Inhalte:</p> <p>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 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

Prüfung

Praktikum Prozessorbau

praktische Prüfung

| | | |
|--|--|--|
| Modul INF-0309: Echtzeitsysteme <i>Real-Time Systems</i> | | 8 ECTS/LP |
| Version 1.5.0 (seit WS19/20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <p>Lernziele/Kompetenzen:</p> <p>Die Vorlesung vermittelt grundlegende und weiterführende Kenntnisse zu Echtzeitsystemen wie sie in nahezu allen eingebetteten Systemen, jedoch speziell in den Bereichen Automobil, Luft-/Raumfahrt und Robotik vorkommen. Die theoretischen Grundlagen werden sich am aktuellen Forschungsstand orientieren und eine weitere Beschäftigung der Studierenden mit dem Thema eingebettete Echtzeitsysteme auf wissenschaftlichen Niveau ermöglichen.</p> <p>Die Vorlesung wird den Studierenden die Fähigkeit vermitteln verschiedene eingebettete Systeme anhand ihrer Echtzeitanforderungen zu unterscheiden und einzuordnen. Die Studierenden lernen aktuelle Methoden zur Validierung des Zeitverhaltens anzuwenden, zu vergleichen, und im Hinblick auf eine mögliche Zertifizierung des Zeitverhaltens kritisch zu analysieren. Dies beinhaltet die Optimierung und Auswahl von Echtzeit-Schedules und deren Verifikation. Die Vorlesung behandelt zudem verschiedene Prozessortypen, und wird näher auf die Besonderheiten von Einkern und Mehrkern-Prozessoren im Echtzeitbereich eingehen. Die Studierenden werden hierbei in die Lage versetzt, Prozessoren anhand ihrer Eignung für Echtzeitsysteme zu klassifizieren und den Einfluss von Entwurfsentscheidungen auf das Echtzeitverhalten und die Analyse des Echtzeitverhaltens zu untersuchen. Der Lehrstoff wird anhand von Fallstudien aus den Bereichen Automobil und Luftfahrt exemplifiziert und von den Studierenden anhand eines einfachen Echtzeitsystems angewendet.</p> <p>Schlüsselqualifikationen:</p> <p>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:</p> <p>Gesamt: 240 Std.</p> <p>30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</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>60 Std. Vorlesung (Präsenzstudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> | | |
| <p>Voraussetzungen:</p> <p>keine</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: Echtzeitsysteme (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch / Englisch</p> <p>SWS: 4</p> | | |

Inhalte:

- WCET Analyse
- Scheduling Analyse
- Programmierung von Echtzeitsystemen
- Prozessoren für Echtzeitsysteme
- Echtzeitbetriebssysteme
- Zertifizierung von Echtzeitsystemen

Literatur:

- Sanjoy Baruah, Marko Bertogna, Giorgio Buttazzo, Multiprocessor Scheduling for Real-Time Systems, Springer, 2015.
- Giorgio Buttazzo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, Springer, 2011.
- Heinz Wörn, Uwe Brinkschulte, Echtzeitsysteme, Springer Verlag, Berlin/Heidelberg, 2005
- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010

Zugeordnete Lehrveranstaltungen:

Echtzeitsysteme (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Echtzeitsysteme sind Computer Systeme, deren Korrektheit nicht nur über das korrekte Ergebnis einer Berechnung, sondern auch über das korrekte Zeitverhalten definiert wird. Vereinfacht gesagt: Wenn ein eingebettetes System zu spät reagiert, ist es genauso unbrauchbar, wie wenn es gar nicht oder falsch reagiert hätte. Ein Airbag zum Beispiel sollte aufgehen, bevor der Kopf des Fahrers auf das Lenkrad knallt. Ein selbstfahrendes Fahrzeug muss ebenso rechtzeitig autonom bremsen, bevor ein Unfall geschehen ist, und eine Kaffeemaschine sollte aufhören Kaffee auszuschenken, bevor die Tasse überläuft. In dieser Vorlesung beschäftigen wir uns mit dem Zeitverhalten eben solcher eingebetteter Echtzeitsysteme. Wir lernen wie wir das Zeitverhalten dieser Systeme beeinflussen und validieren können und wir lernen, welchen Einfluss verschiedene Entwurfsentscheidungen auf das Zeitverhalten haben.

... (weiter siehe Digicampus)

Modulteil: Echtzeitsysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Echtzeitsysteme (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Echtzeitsysteme

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0335: Safety-Critical Systems <i>Safety-Critical Systems</i> | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <p>Lernziele/Kompetenzen:</p> <p>Die Studierenden lernen die Fehlertoleranzanforderungen in eingebetteten Systemen kennen und ordnen diese anhand der zugrundeliegenden Vorschriften und Normen (z.B. ISO 26262) zum Entwurf sicherheitsrelevanter Systeme ein.</p> <p>Sie können die Ursachen und Effekte von Hardware-Fehlern beschreiben und sind in der Lage, unterschiedliche Redundanztechniken zu analysieren und zu bewerten.</p> <p>Weitere Schwerpunkte der Vorlesung sind die Anwendung der stochastischen Grundlagen der Fehlerrechnung, das Analysieren und Modellieren von Rechensystemen mit Zuverlässigkeitsblockdiagrammen und Fehlerbäumen, sowie das Differenzieren unterschiedlicher Redundanzarten und deren Verwendung in verschiedenen Hard- und Software-Techniken zur Fehlererkennung.</p> <p>Außerdem werden Techniken zur Fehlerkorrektur, Fehlerdiagnose und Rekonfiguration vorgestellt und untersucht. Die Studierenden lernen verschiedene Konzepte fehlertolerierender Rechensysteme aus Theorie und Praxis zu klassifizieren und zu vergleichen, wie z.B. Lockstep-Ausführung, Prozess-Redundanz, ECC-Speicher-Implementierungen oder Kontrollflussüberprüfung.</p> <p>In den praktischen Übungen wenden die Studierenden die Methoden der Modellierung und Fehlerrechnung an verschiedenen Aufgaben an, arbeiten ihre Lösungen aus und vergleichen auf qualitativer Ebene verschiedene Fehlertoleranzkonzepte aus Forschung und Praxis.</p> <p>Schlüsselqualifikationen: Analytisch-methodische Kompetenz in der Analyse fehlertolerierender Rechensysteme, Abwägung von Lösungsansätzen, Verständliche Präsentation von Lösungen von Übungsaufgaben, Fähigkeit, vorhandenes Wissen selbstständig zu erweitern, Selbstreflexion</p> | | |
| <p>Arbeitsaufwand:</p> <p>Gesamt: 150 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>60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>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 | | |
| <p>Modulteil: Safety-Critical Systems (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p> | | |

Inhalte:

Die Vorlesung betrachtet Methoden zur Einhaltung der funktionalen Sicherheit in Rechensystemen, sowie deren Entwurf und Analyse. 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. Verschiedene Methoden zur Fehlerinjektion, die zur Bewertung von fehlertolerierenden Systemen notwendig sind, werden kurz besprochen.

Literatur:

- D. Sorin: Fault Tolerant Computer Architecture, Morgan and Claypool, 2009
- S. Mukherjee: Architecture Design for Soft Errors, Morgan Kaufmann, 2008
- I. Koren, C.M. Krishna: Fault-Tolerant Systems, Morgan Kaufmann, 2007

Modulteil: Safety-Critical Systems (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Safety-Critical Systems

Klausur / Prüfungsdauer: 60 Minuten

| | | |
|---|---|---|
| Modul INF-0337: Seminar Embedded Systems (Master) <i>Seminar Embedded Systems (Master)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <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 Embedded Systems 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.</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 unter unterschiedlichen Gesichtspunkten</p> | | |
| <p>Arbeitsaufwand:</p> <p>Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p> | | |
| <p>Voraussetzungen:</p> <p>keine</p> | | |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Embedded Systems (Master) | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| <p>Inhalte:</p> <p>Im Seminar werden Themen aus dem Bereich Embedded Systems 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 Embedded Systems (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

Alle wichtigen Informationen zum Seminar finden Sie im Seminar-Wiki unter es-augsburg.de/seminar/.

Prüfung

Seminar Embedded Systems (Master)

Schriftlich-Mündliche Prüfung

| | | |
|--|---|---|
| Modul INF-0340: Projektmodul Embedded Systems <i>Project Module Embedded Systems</i> | | 10 ECTS/LP |
| Version 1.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet der Embedded Systems 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 |
| <p>Modulteil: Projektmodul Embedded Systems Lehrformen: Praktikum Sprache: Deutsch SWS: 1</p> |
| <p>Inhalte: Autonome Mitarbeit an aktuelle Forschungsthemen.</p> |
| <p>Literatur: wissenschaftliche Papiere, Handbücher</p> |
| <p>Zugeordnete Lehrveranstaltungen: Oberseminar Embedded Systems <i>*Veranstaltung wird online/digital abgehalten.*</i> Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren direkt einen Mitarbeiter, dessen Forschung Sie interessiert: www.es-augsburg.de Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte.</p> |

Prüfung

Projektmodul Embedded Systems

praktische Prüfung

| | | |
|---|---|--|
| Modul INF-0367: Advanced Machine Learning and Computer Vision | | 5 ECTS/LP |
| Version 1.0.0 (seit WS20/21) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart | | |
| <p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul besitzen die Studierenden vertiefte fortgeschrittene Kenntnisse des maschinellen Lernens (Support Vector Machines und tiefe neuronale Netze und deren Grundbausteine) und des maschinellen Sehens (tiefe neuronale Netzarchitekturen und Systeme) und können diese anwenden. Sie sind in der Lage, wissenschaftlich komplexe Verfahren auf dem Gebiet der Bild-, Text-, Video- und Signalverarbeitung zu analysieren, zu verstehen und programmatisch umzusetzen, sowie die erlernten Prinzipien auf neue Fragestellungen geeignet anzuwenden. Sie entwickeln Fertigkeiten zum logischen, analytischen und konzeptionellen Denken im Bereich des maschinellen Lernens und Sehens.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Logik; kritisches Lesen und Analysieren von wissenschaftlichen Veröffentlichungen; 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. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 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)</p> | | |
| <p>Voraussetzungen: Kenntnisse in maschinellern Lernen und maschinellern Sehen (Master-Vorlesung INF-0092 "Multimedia II" bzw. INF-0316 "Machine Learning and Computer Vision")</p> | | <p>ECTS/LP-Bedingungen: Bestehen der Modulprüfung</p> |
| <p>Angebotshäufigkeit: jedes Wintersemester</p> | <p>Empfohlenes Fachsemester: ab dem 2.</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: Advanced Machine Learning and Computer Vision (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p> | | |
| <p>Inhalte: Die Vorlesung gibt einen vertieften Einblick in alle Aspekte des maschinellen Lernens und des maschinellen Sehens. Die erlernten Konzepte werden in den Übungen anhand von erfolgreichen Beispielen aus der Praxis ausprobiert, geübt, analysiert und bewertet. Die Inhalte der Vorlesung umfassen: Support Vector Machines, Grundbausteine von tiefen Neuronalen Netzen (Layerstrukturen, Normalisierung, Attention-Mechanismen), sowie aktuelle Referenzarchitekturen und -systeme für Bild-, Text-, Videoverarbeitung und deren Kombination mit weiteren Sensorsignalen.</p> | | |
| <p>Literatur: Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.</p> | | |
| <p>Zugeordnete Lehrveranstaltungen:</p> | | |

Advanced Machine Learning and Computer Vision (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Modulteil: Advanced Machine Learning and Computer Vision (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Advanced Machine Learning and Computer Vision (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Advanced Machine Learning and Computer Vision (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

| | | |
|--|---|---|
| Modul INF-0368: Embedded Hardware <i>Embedded Hardware</i> | | 5 ECTS/LP |
| Version 1.0.0 (seit WS20/21) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <p>Lernziele/Kompetenzen:</p> <p>Die Vorlesung vermittelt grundlegende und weiterführende Kenntnisse zu Hardwarekomponenten (z.B. GPIO, Hardware-Timer, ...) wie sie in nahezu allen eingebetteten Systemen vorkommen. Außerdem werden die Funktionsprinzipien von verschiedenen Sensoren und Aktuatoren vorgestellt. Ein besonderer Fokus liegt dabei auf deren Einsatz zur Interaktion mit der physikalischen Welt und auf Besonderheiten bei deren Anwendung. Die theoretischen Grundlagen werden sich am aktuellen Forschungsstand orientieren und eine weitere Beschäftigung der Studierenden mit dem Thema eingebetteter Hardware auf wissenschaftlichen Niveau ermöglichen.</p> <p>Die Vorlesung wird den Studierenden die Fähigkeit vermitteln verschiedene eingebettete Systeme anhand ihrer Hardwareanforderungen zu unterscheiden und einzuordnen. Die Studierenden lernen aktuelle Methoden zur Programmierung der Komponenten anzuwenden, zu vergleichen, und im Hinblick auf eine effiziente Nutzung kritisch zu analysieren. Dies beinhaltet die Auswahl und Beurteilung von für das System relevanten Hardwarekomponenten, sowie die Entwicklung einer für ein bestehendes Problem optimierten Konfiguration der Komponenten. Die Vorlesung behandelt zudem verschiedene Sensoren und Aktuatoren, und wird näher auf die Umwandlung zwischen einer analogen, physikalischen Größe und einer digitalen, elektrischen Größe eingehen. Die Studierenden werden hierbei in die Lage versetzt, physikalische Größen anhand ihrer Eigenschaften zu klassifizieren und eine Entscheidung über die für die Umwandlung benötigten Sensoren, Aktuatoren und Hardwarekomponenten zu treffen.</p> <p>Der Lehrstoff wird anhand von Fallstudien z.B. aus den Bereich Automobil exemplifiziert und von den Studierenden anhand eines einfachen eingebetteten Systems angewendet.</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:</p> <p>Gesamt: 150 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>60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| <p>Modulteil: Embedded Hardware (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch / Englisch</p> <p>SWS: 2</p> | | |

Inhalte:

- Modellierung von Sensoren und Aktuatoren
- Grundlagen in Elektronik
- Hardwarekomponenten von eingebetteten Prozessoren und SoC
- Serielle Schnittstellen
- Sensoren und Eingabegeräte
- Aktuatoren und Ausgabegeräte

Literatur:

- Yifeng Zhu, Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C, E-Man Press LLC, 2017
- Edward Ashford Lee, Sanjit Arunkumar Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, MIT Press, 2017
- Elecia White, Making Embedded Systems, O'Reilly Media, 2012
- John Catsoulis, Designing Embedded Hardware, O'Reilly Media, 2005
- Rüdiger R. Asche, Embedded Controller: Grundlagen und praktische Umsetzung für industrielle Anwendungen, Springer, 2016
- Hermann Winner, Stephan Hakuli, Felix Lotz, Christina Singer Hrsg., Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort, Springer, 2015
- Bernhard Grimm, Gregor Häberle, Heinz Häberle, u.a., Fachkunde Industrieelektronik und Informationstechnik, Europa Lehrmittel, 2003

Zugeordnete Lehrveranstaltungen:

Embedded Hardware (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Eingebettete Systeme sind Computer Systeme, deren vorrangige Aufgabe es ist die umgebende physikalische und technische Welt zu erfassen bzw. zu beeinflussen. Diese kann beispielsweise durch regelmäßiges Messen physikalischer Größen oder durch die Steuerung bzw. Regelung von technischen Anlagen geschehen. Um diese Aufgaben zu bewerkstelligen benötigen eingebettete Prozessoren diverse Hardwarekomponenten, die es ihnen ermöglicht mit der Außenwelt zu interagieren. Zusätzlich werden Sensoren und Aktuatoren benötigt. Sie übernehmen die Umwandlung von physikalischen Größen in elektrische Signale und umgekehrt. In dieser Vorlesung beschäftigen wir uns sowohl mit den beschriebenen Hardwarekomponenten als auch mit verschiedenen Sensoren und Aktuatoren. Wir lernen wie ein Prozessor mit Hilfe dieser Komponenten mit seiner Umgebung interagieren kann und welchen Einfluss die Komponenten auf die Wandlung von physikalischen Größen zu einem digitalen Wert (und umgekehrt) haben kann.
 ... (weiter siehe Digicampus)

Modulteil: Embedded Hardware (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Embedded Hardware (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Embedded Hardware

Klausur / Prüfungsdauer: 60 Minuten

| | | |
|--|---|---|
| Modul INF-0037: Praktikum Automotive Software Engineering <i>Practical Module Automotive Software Engineering</i> | | 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 Fahrerassistenzsystemen mit aktuellen Methoden und Tools der modellbasierten Entwicklung eingebetteter Systeme. Die Studierenden erlangen tiefere 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> | | |
| <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: 6</p> | <p>Wiederholbarkeit: siehe PO des Studiengangs</p> | |
| <p>Modulteile</p> | | |
| <p>Modulteil: Praktikum Automotive Software Engineering (MA) Lehrformen: Praktikum Sprache: Deutsch Angebotshäufigkeit: jedes Wintersemester 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

Prüfung

Praktikum Automotive Software Engineering (MA)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

| | | |
|---|---|---|
| Modul INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme <i>Project Module Software Methodologies for Distributed Systems</i> | | 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 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. 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 <i>*Veranstaltung wird online/digital abgehalten.*</i> | | |
| Prüfung | | |
| Projektabnahme, Vortrag, Abschlussbericht Praktikum | | |

| | | |
|---|---|---|
| Modul INF-0066: Organic Computing II <i>Organic Computing II</i> | | 5 ECTS/LP |
| Version 1.3.0 (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 <i>Peer-to-Peer and Cloud Computing</i> | | 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

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

Lehrformen: Übung

Sprache: Deutsch

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

SWS: 2

Prüfung

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

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

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

| | | |
|---|---|---|
| Modul INF-0069: Weiterführende Betriebssystemkonzepte <i>Advanced Operating System Concepts</i> | | 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-0070: Seminar Organic Computing <i>Seminar Organic Computing</i> | | 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 | |
| Modulteile | | |
| 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)

Veranstaltung wird online/digital abgehalten.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

| | | |
|---|---|---|
| Modul INF-0071: Seminar Naturalogische Algorithmen und Multiagentensysteme <i>Seminar Nature inspired Algorithms and Multi Agent Systems</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| Lernziele/Kompetenzen: 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. 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. | | |
| Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) | | |
| 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 Naturalogische Algorithmen und Multiagentensysteme Lehrformen: Seminar Sprache: Deutsch SWS: 2 | | |

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-0072: Projektmodul Organic Computing <i>Project Module Organic Computing</i> | | 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 | | |
| <p>Modulteil: Projektmodul Organic Computing Lehrformen: Praktikum Sprache: Deutsch SWS: 1</p> | | |
| Inhalte: Mitarbeit an aktuellen Forschungsthemen. | | |
| <p>Literatur: In Abhängigkeit vom zu bearbeitenden Thema:</p> <ul style="list-style-type: none"> • Paper • Buch • Handbuch | | |
| <p>Zugeordnete Lehrveranstaltungen: Oberseminar Organic Computing <i>*Veranstaltung wird online/digital abgehalten.*</i> Weiterführende Themen aus dem Bereich Organic Computing</p> | | |

Prüfung

Vortrag und Abschlussbericht

Praktikum

| | | |
|--|--|--|
| Modul INF-0088: Bayesian Networks <i>Bayesian Networks</i> | | 5 ECTS/LP |
| Version 1.0.0 (seit WS15/16) 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 <i>Multimedia II: Machine Learning and Computer Vision</i> | | 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 zum 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 | | |
| Modulteil: Multimedia II: Machine Learning and Computer Vision (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 4 | | |
| <p>Inhalte: Die Vorlesung gibt einen guten Überblick über alle Aspekte des maschinellen Lernens und der maschinellen Extraktion von Informationen aus Multimediadaten (z.B. "Google Image Search", "Google Goggles"). Die erlernten Konzepte werden in den Übungen anhand von erfolgreichen Beispielen aus der Praxis ausprobiert, geübt, analysiert und bewertet. Zum Ende des Semesters werden fortgeschrittene Themen wie Objektdetektion und Objekterkennung von Gesichtern und Menschen behandelt. Die Inhalte der Vorlesung umfassen: Machine Learning (Decision Tree Learning, Artificial Neural Networks, Bayesian Learning, Discrete Adaboost), Data Reduction (Quantization (K-Means Clustering, Affinity Propagation), Dimensionality Reduction Techniques (PCA, NMF, Random Projection, MDS)) und Image Processing & Computer Vision (Salient Feature Points and Feature Descriptors, Object Detection (Face/Car/People Detection), Object Recognition (Face Recognition), Image Search with pLSA)</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 <i>Probabilistic Robotics</i> | | 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: 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: Probabilistic Robotics (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch Angebotshäufigkeit: jedes Wintersemester SWS: 2</p> | | |

Inhalte:

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.

Modulteil: Probabilistic Robotics (Übung)

Lehrformen: Übung

Sprache: Deutsch

Angebotshäufigkeit: jedes Wintersemester

SWS: 2

Prüfung

Probabilistic Robotics (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

| | | |
|---|--|--|
| Modul INF-0129: Softwaretechnik II <i>Software Engineering II</i> | | 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> | | |
| Voraussetzungen: | | |
| Programmierkenntnisse in Java (empfohlen) Modul Softwaretechnik (INF-0120) - empfohlen | | |
| 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:****Agile Softwareentwicklung:**

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

Refactoring

- Code Smells
- Prinzipien des objektorientierten Designs
- Wichtige Refactorings

Testen

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

Requirements Engineering

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

Literatur:

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

Zugeordnete Lehrveranstaltungen:**Softwaretechnik 2 (Vorlesung)**

Veranstaltung wird als Hybrid/gemischt abgehalten.

Die Veranstaltung findet im Wesentlichen digital statt. Vorlesungsinhalte werden als Screencast zum Download angeboten. Zusätzlich gibt es pro Woche höchstens einen synchronen Termin, bei dem Fragen zum Stoff gestellt werden können und die Übungen behandelt werden. Für den synchronen Termin ist im Stundenplan Dienstag ab 16 Uhr vorgesehen. Falls es die Corona-Situation und die Teilnehmerzahl erlauben, gibt es am Anfang des neuen Jahres noch zwei Präsenztermine, in denen Stoff praktisch erarbeitet wird.

Modulteil: Softwaretechnik II (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Softwaretechnik 2 (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Softwaretechnik II Klausur

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0131: Software- und Systemsicherheit <i>Software and Systems Security</i> | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| Lernziele/Kompetenzen: Fachbezogene Kompetenzen: 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. Sie können Bedrohungsanalyse durchführen, kryptographische Protokolle entwickeln, Chipkarten programmieren, sicherheitskritische Systeme in Teams entwerfen, und ihre Ergebnisse dokumentieren. Sie können systematisch Bedrohungen für Softwaresysteme analysieren und deren Risiken bewerten. Sie können einen modellgetriebenen Entwicklungsprozess für sicherheitskritische Systeme anwenden. Sie sind in der Lage, bedeutende technische Entwicklungen zu erkennen und fachliche Lösungskonzepte in Programme umzusetzen. Sie haben Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen. Schlüsselqualifikationen: <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 | | |
| 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: 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 | | |
| Modulteil: Software- und Systemsicherheit (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2 | | |

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 <i>Self-organizing, adaptive systems</i> | | 8 ECTS/LP |
| Version 3.0.0 (seit WS20/21) 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)

Veranstaltung wird online/digital abgehalten.

"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" sowie flexible Produktion (<https://www.uni-augsburg.de/de/fakultaet/fai/isse/projects/soproduction/>) und Planung und Selbstorganisation in mobilen Multi-Roboter-Systemen (<https://www.uni-augsburg.de/de/fakultaet/fai/isse/projects/combo/>) aus.

Modulteil: Selbstorganisierende, adaptive Systeme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Selbst-organisierende, adaptive Systeme (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Selbstorganisierende, adaptive Systeme

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

Beschreibung:

Ausnahmefall WiSe 2020/2021: Die Prüfung findet als Klausur statt.

| | | |
|---|---|---|
| Modul INF-0136: Seminar Software- und Systems Engineering (Master) <i>Seminar Software- und Systems Engineering (Master)</i> | | 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 | |

| |
|---|
| Moduleile |
| Moduleil: Seminar Software- und Systems Engineering (Master) Lehrformen: Seminar Sprache: Deutsch / Englisch Angebotshäufigkeit: jedes Sommersemester SWS: 2 |
| 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. |
| Literatur: Abhängig von den konkreten Themen des Seminars |
| Zugeordnete Lehrveranstaltungen: Seminar zu IT-Sicherheit (Seminar) <i>*Veranstaltung wird online/digital abgehalten.*</i> Es werden maximal die 10 unten aufgeführten Themen nach dem Prinzip First Come, First Served vergeben. Um am Seminar teilzunehmen, schicken Sie eine Mail mit ZWEI Themenwünschen an philip.lenzen@informatik.uni-augsburg.de. Folgende Seminarthemen werden angeboten: Themenblock 1: Exploits 1. Angriffe auf Signaturen von PDF-Dokumenten Das "Portable Document Format" (PDF) ist de-facto Standard für elektronische Dokumente, unter anderem für elektronische Verträge und Rechnungen im Staats- und Finanzwesen. Zur Sicherung der Authentizität und Integrität von PDF-Dokumenten werden Digitale Signaturen eingesetzt. Allerdings ist eine Vielzahl an PDF-Viewern verwundbar hinsichtlich neuartiger Angriffe auf Signaturen, die Forscher der Ruhr-Universität Bochum entdeckt haben. Diese sollen im Rahmen dieser Arbeit näher untersucht werden. 2. Automatische Heap-basierte Exploit-Generierung Typische Ansätze der automatisierten Generierung von Exploits (Befehlsketten, die aufgrund von Speicherkorruption be ... (weiter siehe Digicampus) |
| Prüfung Seminar Software- und Systems Engineering (Master) Seminar / Prüfungsdauer: 45 Minuten Bearbeitungsfrist: 3 Monate |

| | | |
|---|---|---|
| Modul INF-0137: Projektmodul Software- und Systems Engineering <i>Project Module Software- and Systems Engineering</i> | | 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 | | |
| Modulteil: Projektmodul Software- und Systems Engineering | | |
| Lehrformen: Praktikum | | |
| Sprache: Deutsch / Englisch | | |
| SWS: 1 | | |
| Inhalte: Mitarbeit an aktuellen Forschungsthemen des Lehrstuhls | | |
| Literatur: Abhängig von dem konkreten Projekt: Wissenschaftliche Papiere, Dokumentation | | |
| Zugeordnete Lehrveranstaltungen: Oberseminar Software- und Systems Engineering | | |

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen am Lehrstuhl für Softwaretechnik zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren Sie direkt einen Mitarbeiter, dessen Forschung Sie interessiert: <https://www.uni-augsburg.de/de/fakultaet/fai/isse/prof/swtse/teaching/students/> Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Prüfung

Projektmodul Software- und Systems Engineering Projektabnahme

praktische Prüfung / Bearbeitungsfrist: 2 Monate

| | | |
|--|---|---|
| Modul INF-0147: Prozessorarchitektur <i>Processor Architecture</i> | | 5 ECTS/LP |
| Version 2.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer Dr. Martin Frieb | | |
| <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. 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Übung (Präsenzstudium)</p> | | |
| Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen | | |
| Angebotshäufigkeit: unregelmäßig | 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 superskalärer 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

Modulteil: Prozessorarchitektur (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Prozessorarchitektur

Klausur / Prüfungsdauer: 60 Minuten

| | | |
|--|---|---|
| Modul INF-0149: Praktikum Eingebettete Systeme <i>Practical Module Embedded Systems</i> | | 5 ECTS/LP |
| Version 2.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <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. 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Praktikum (Präsenzstudium)</p> | | |
| Voraussetzungen: Kenntnisse in C. | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Praktikum Eingebettete Systeme | | |
| Lehrformen: Praktikum | | |
| Sprache: Deutsch | | |
| SWS: 4 | | |
| <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.</p> <p>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

Praktikum Eingebettete Systeme

praktische Prüfung

| | | |
|---|---|---|
| Modul INF-0150: Hardware-Entwurf <i>Hardware Design</i> | | 8 ECTS/LP |
| Version 2.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer Dr. Martin Frieb | | |
| <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>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: Modul Systemnahe Informatik (INF-0138) - empfohlen | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |

| |
|---|
| Modulteile |
| <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

Hardware-Entwurf

praktische Prüfung

Beschreibung:

Erfolgreiche Praktikumsteilnahme, Projektvorstellung am Ende des Semesters

| | | |
|--|---|---|
| Modul INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung <i>Practical Module Multimodal Real Time Signal Processing</i> | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André | | |
| 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. | | |
| 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 | | |
| Arbeitsaufwand: Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) | | |
| Voraussetzungen: Programmiererfahrung Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia Grundlagen II (INF-0166) - empfohlen | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Praktikum Multimodale Echtzeitsignalverarbeitung Lehrformen: Praktikum Sprache: Deutsch SWS: 6 | | |
| Inhalte: Die konkrete Aufgabenstellung aus dem weiten Gebiet der multimodalen Echtzeitsignalverarbeitung wird jedes Jahr neu entworfen. | | |
| Literatur: Literaturhinweise werden je nach Thema zu Beginn des Semesters bekanntgegeben. | | |
| Prüfung Praktikum Multimodale Echtzeitsignalverarbeitung praktische Prüfung | | |

| | | |
|--|---|---|
| Modul INF-0189: Qualitätssicherung im Software Engineering <i>Quality Assurance in Software Engineering</i> | | 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-0216: Vertiefte Multicore-Programmierung <i>Advanced Multicore Programming</i> | | 8 ECTS/LP |
| Version 2.0.0 (seit SoSe20 bis WS20/21) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <p>Lernziele/Kompetenzen:</p> <p>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 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:</p> <p>Gesamt: 240 Std.</p> <p>60 Std. Übung (Präsenzstudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>120 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> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Vertiefte Multicore-Programmierung (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| 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. | | |

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 (bung)

Lehrformen: bung

Sprache: Deutsch

SWS: 4

Inhalte:

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

Prufung

Vertiefte Multicore-Programmierung

Portfolioprufung

Beschreibung:

Projektvorstellung und -abnahme, Fragen zu Vorlesung und bung

| | | |
|---|---|---|
| Modul INF-0217: Praktikum Autonomes Fahren <i>Practical Module Autonomous Driving</i> | | 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> | | |
| <p>Angebotshäufigkeit: jedes Sommersemester</p> | <p>Empfohlenes Fachsemester: ab dem 1.</p> | <p>Minimale Dauer des Moduls: 1 Semester</p> |
| <p>SWS: 10</p> | <p>Wiederholbarkeit: siehe PO des Studiengangs</p> | |
| <p>Moduleile</p> | | |
| <p>Moduleil: 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-0232: Seminar Medical Information Sciences (MA) <i>Seminar Medical Information Sciences (MA)</i> | | 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> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Medical Information Sciences (Seminar) | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| 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. | | |
| Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt. | | |
| Zugeordnete Lehrveranstaltungen: | | |

Seminar Medical Information Sciences (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

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

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

| | | |
|---|---|---|
| Modul INF-0233: Industrierobotik <i>Industrial Robotics</i> | | 8 ECTS/LP |
| Version 2.2.0 (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 Wintersemester | 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)

Veranstaltung wird als Hybrid/gemischt abgehalten.

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)

Veranstaltung wird als Hybrid/gemischt abgehalten.

Prüfung

Industrierobotik

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 3 Monate

Beschreibung:

Ausnahmefall WiSe 2020/2021: Die Prüfung findet als Portfolioprüfung statt.

| | | |
|---|---|---|
| Modul INF-0234: Flugrobotik <i>Aerial Robotics</i> | | 5 ECTS/LP |
| Version 1.3.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| <p>Lernziele/Kompetenzen: Die Studierenden kennen die Grundlagen fliegender Roboter, sind in der Lage, diese zu programmieren, und können die Eignung verschiedener Vorgehensweisen dafür bewerten. Sie verstehen den Aufbau von Flugrobotern sowie die Funktionsweise oft verwendeter Sensorik. Sie können fachliche Lösungskonzepte in Software umsetzen und dabei Entwurfsalternativen bewerten und anwenden. Die Studierenden sind in der Lage für praxisrelevante Aufgabenstellungen die Eignung fliegender Roboter zu bewerten. Sie haben die Fähigkeit zum analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikation:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenz • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten | | |
| <p>Bemerkung: "Grundlagen der Robotik" als Vorkenntnis empfohlen</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: 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 | |
| Moduleile | | |
| <p>Modulteil: Flugrobotik (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p> | | |

Inhalte:

Fliegende Roboter, wie zum Beispiel Multicopter, sind immer häufiger im Alltag anzutreffen. Durch Fortschritte im Bereich von Mikrocontrollern, leistungsfähigen Elektromotoren und der Akku-Technologie sind fliegende Roboter für viele Anwendungsbereiche interessant geworden und werden in Zukunft noch viel stärker an Bedeutung gewinnen.

In dieser Veranstaltung liegt der Schwerpunkt auf der Entwicklung von Software für fliegende Robotersysteme. Dazu werden Grundlagen in folgenden Bereichen vermittelt:

- Aufbau und Funktionsweise von Multicoptern
- Aufbau von Flugroboter-Steuerungen
- integrierte Sensorik und deren Funktionsweise
- Verarbeitung verrauschter Sensordaten
- Grundlagen räumlicher Beschreibungsformalismen
- Steuerungsmodi von Flugrobotern und autonome Grundfähigkeiten
- Rechtliche Grundlagen für den Einsatz von Flugrobotern

Auf Basis dieser Grundlagen werden verschiedene praktische Aufgabenstellungen prototypisch umgesetzt. Die Softwareentwicklung erfolgt in objektorientierten Programmiersprachen. Die Prototypen werden im Rahmen der begleitenden Übung evaluiert und optimiert.

Literatur:

- Probabilistic Robotics. Sebastian Thrun, Wolfram Burgard and Dieter Fox. MIT Press, 2005.
- Planning Algorithms. Steven M. LaValle. Cambridge University Press, 2006.

Modulteil: Flugrobotik (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Flugrobotik

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 3 Monate

| | | |
|---|---|---|
| Modul INF-0235: Software für Industrie 4.0 <i>Software for Industry 4.0</i> | | 5 ECTS/LP |
| Version 1.2.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| <p>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.</p> <p>Schlüsselqualifikationen:</p> <ul style="list-style-type: none"> • Analytisch-methodische Kompetenzen • Abwägen von Lösungsansätzen • Erwerb von Abstraktionsfähigkeiten | | |
| <p>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)</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: Software für Industrie 4.0 (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p> | | |
| <p>Inhalte: Die Vorlesung behandelt aktuelle Themen im Kontext von Industrie 4.0. Dazu zählen folgende Inhalte:</p> <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 | | |
| <p>Literatur:</p> <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 <i>Collaborative Robotics</i> | | 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> | | |
| Voraussetzungen: keine | | ECTS/LP-Bedingungen: Bestehen der Modulprüfung |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Moduleile | | |
| <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 vernetzen 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)**

Veranstaltung wird als Hybrid/gemischt abgehalten.

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)**

Veranstaltung wird als Hybrid/gemischt abgehalten.

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-0251: Seminar Artificial Intelligence <i>Seminar Artificial Intelligence</i> | | 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: 4.0 | | |

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

Prüfung

Seminar Artificial Intelligence

Schriftlich-Mündliche Prüfung

| | | |
|---|---|--|
| Modul INF-0291: Praktikum Selbstlernende Systeme <i>Practical Module Self-Learning Systems</i> | | 8 ECTS/LP |
| Version 1.2.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| 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. | | |
| Schlüsselqualifikationen: <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 | | |
| Arbeitsaufwand: Gesamt: 240 Std. 225 Std. Praktikum (Selbststudium) 15 Std. Teilnahme an Lehrveranstaltungen (Präsenzstudium) | | |
| Voraussetzungen: Empfohlen: die erfolgreiche Teilnahme an der Vorlesung OC2, Programmiererfahrung, Teamfähigkeit Modul Organic Computing II (INF-0066) - empfohlen | | ECTS/LP-Bedingungen: Bestehen der Modulprüfung |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: ab dem 2. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Praktikum Selbstlernende Systeme Lehrformen: Praktikum Dozenten: Prof. Dr. Jörg Hähner Sprache: Deutsch SWS: 2 ECTS/LP: 8.0 | | |

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)

Veranstaltung wird online/digital abgehalten.

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-0297: Praktikum Prozessorbau <i>Practical Module Processor Design</i> | | 5 ECTS/LP |
| Version 2.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer Dr. Martin Frieb | | |
| <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>Arbeitsaufwand: Gesamt: 150 Std. 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Praktikum (Präsenzstudium)</p> | | |
| <p>Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen Modul Prozessorarchitektur (INF-0147) - empfohlen</p> | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Moduleile | | |
| <p>Modulteil: Praktikum Prozessorbau Lehrformen: Praktikum Sprache: Deutsch SWS: 4</p> | | |
| <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 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

Prüfung

Praktikum Prozessorbau

praktische Prüfung

| | | |
|---|--|--|
| Modul INF-0307: Modellbasierte Entwicklung und Analyse von Software Systemen <i>Model-Based Development and Analysis of Software Systems</i> | | 6 ECTS/LP |
| Version 1.1.0 (seit SoSe19) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer | | |
| <p>Lernziele/Kompetenzen: Modellbasierte Entwicklung und Analyse von Software Systemen 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. 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> | | |
| Voraussetzungen: Frühere Veranstaltung "Modellgetriebene Softwareentwicklung" darf nicht belegt worden sein wegen Überschneidungen. | | |
| Angebotshäufigkeit: jedes Sommersemester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 5 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Modellbasierte Entwicklung und Analyse von Software Systemen (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 3 | | |
| Inhalte: Modellbasierte Entwicklung und Analyse von Software Systemen befasst sich mit der Effizienzsteigerung in der Softwareherstellung durch Automatisierung und Wiederverwendung. Dabei werden Infrastrukturcode, Subsysteme, Konfigurationen oder ganze Anwendungen aus Modellen generiert. | | |

Literatur:

- 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: Modellbasierte Entwicklung und Analyse von Software Systemen (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Modellbasierte Entwicklung und Analyse von Software Systemen

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

| | | |
|--|---|---|
| Modul INF-0309: Echtzeitsysteme <i>Real-Time Systems</i> | | 8 ECTS/LP |
| Version 1.5.0 (seit WS19/20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <p>Lernziele/Kompetenzen:</p> <p>Die Vorlesung vermittelt grundlegende und weiterführende Kenntnisse zu Echtzeitsystemen wie sie in nahezu allen eingebetteten Systemen, jedoch speziell in den Bereichen Automobil, Luft-/Raumfahrt und Robotik vorkommen. Die theoretischen Grundlagen werden sich am aktuellen Forschungsstand orientieren und eine weitere Beschäftigung der Studierenden mit dem Thema eingebettete Echtzeitsysteme auf wissenschaftlichen Niveau ermöglichen.</p> <p>Die Vorlesung wird den Studierenden die Fähigkeit vermitteln verschiedene eingebettete Systeme anhand ihrer Echtzeitanforderungen zu unterscheiden und einzuordnen. Die Studierenden lernen aktuelle Methoden zur Validierung des Zeitverhaltens anzuwenden, zu vergleichen, und im Hinblick auf eine mögliche Zertifizierung des Zeitverhaltens kritisch zu analysieren. Dies beinhaltet die Optimierung und Auswahl von Echtzeit-Schedules und deren Verifikation. Die Vorlesung behandelt zudem verschiedene Prozessortypen, und wird näher auf die Besonderheiten von Einkern und Mehrkern-Prozessoren im Echtzeitbereich eingehen. Die Studierenden werden hierbei in die Lage versetzt, Prozessoren anhand ihrer Eignung für Echtzeitsysteme zu klassifizieren und den Einfluss von Entwurfsentscheidungen auf das Echtzeitverhalten und die Analyse des Echtzeitverhaltens zu untersuchen. Der Lehrstoff wird anhand von Fallstudien aus den Bereichen Automobil und Luftfahrt exemplifiziert und von den Studierenden anhand eines einfachen Echtzeitsystems angewendet.</p> <p>Schlüsselqualifikationen:</p> <p>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:</p> <p>Gesamt: 240 Std.</p> <p>30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium)</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>60 Std. Vorlesung (Präsenzstudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> | | |
| <p>Voraussetzungen:</p> <p>keine</p> | | |
| <p>Angebotshäufigkeit: jedes Wintersemester</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: Echtzeitsysteme (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch / Englisch</p> <p>SWS: 4</p> | | |

Inhalte:

- WCET Analyse
- Scheduling Analyse
- Programmierung von Echtzeitsystemen
- Prozessoren für Echtzeitsysteme
- Echtzeitbetriebssysteme
- Zertifizierung von Echtzeitsystemen

Literatur:

- Sanjoy Baruah, Marko Bertogna, Giorgio Buttazzo, Multiprocessor Scheduling for Real-Time Systems, Springer, 2015.
- Giorgio Buttazzo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, Springer, 2011.
- Heinz Wörn, Uwe Brinkschulte, Echtzeitsysteme, Springer Verlag, Berlin/Heidelberg, 2005
- Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010

Zugeordnete Lehrveranstaltungen:

Echtzeitsysteme (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Echtzeitsysteme sind Computer Systeme, deren Korrektheit nicht nur über das korrekte Ergebnis einer Berechnung, sondern auch über das korrekte Zeitverhalten definiert wird. Vereinfacht gesagt: Wenn ein eingebettetes System zu spät reagiert, ist es genauso unbrauchbar, wie wenn es gar nicht oder falsch reagiert hätte. Ein Airbag zum Beispiel sollte aufgehen, bevor der Kopf des Fahrers auf das Lenkrad knallt. Ein selbstfahrendes Fahrzeug muss ebenso rechtzeitig autonom bremsen, bevor ein Unfall geschehen ist, und eine Kaffeemaschine sollte aufhören Kaffee auszuschenken, bevor die Tasse überläuft. In dieser Vorlesung beschäftigen wir uns mit dem Zeitverhalten eben solcher eingebetteter Echtzeitsysteme. Wir lernen wie wir das Zeitverhalten dieser Systeme beeinflussen und validieren können und wir lernen, welchen Einfluss verschiedene Entwurfsentscheidungen auf das Zeitverhalten haben.

... (weiter siehe Digicampus)

Modulteil: Echtzeitsysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Echtzeitsysteme (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Echtzeitsysteme

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0335: Safety-Critical Systems <i>Safety-Critical Systems</i> | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <p>Lernziele/Kompetenzen:</p> <p>Die Studierenden lernen die Fehlertoleranzanforderungen in eingebetteten Systemen kennen und ordnen diese anhand der zugrundeliegenden Vorschriften und Normen (z.B. ISO 26262) zum Entwurf sicherheitsrelevanter Systeme ein.</p> <p>Sie können die Ursachen und Effekte von Hardware-Fehlern beschreiben und sind in der Lage, unterschiedliche Redundanztechniken zu analysieren und zu bewerten.</p> <p>Weitere Schwerpunkte der Vorlesung sind die Anwendung der stochastischen Grundlagen der Fehlerrechnung, das Analysieren und Modellieren von Rechensystemen mit Zuverlässigkeitsblockdiagrammen und Fehlerbäumen, sowie das Differenzieren unterschiedlicher Redundanzarten und deren Verwendung in verschiedenen Hard- und Software-Techniken zur Fehlererkennung.</p> <p>Außerdem werden Techniken zur Fehlerkorrektur, Fehlerdiagnose und Rekonfiguration vorgestellt und untersucht. Die Studierenden lernen verschiedene Konzepte fehlertolerierender Rechensysteme aus Theorie und Praxis zu klassifizieren und zu vergleichen, wie z.B. Lockstep-Ausführung, Prozess-Redundanz, ECC-Speicher-Implementierungen oder Kontrollflussüberprüfung.</p> <p>In den praktischen Übungen wenden die Studierenden die Methoden der Modellierung und Fehlerrechnung an verschiedenen Aufgaben an, arbeiten ihre Lösungen aus und vergleichen auf qualitativer Ebene verschiedene Fehlertoleranzkonzepte aus Forschung und Praxis.</p> <p>Schlüsselqualifikationen: Analytisch-methodische Kompetenz in der Analyse fehlertolerierender Rechensysteme, Abwägung von Lösungsansätzen, Verständliche Präsentation von Lösungen von Übungsaufgaben, Fähigkeit, vorhandenes Wissen selbstständig zu erweitern, Selbstreflexion</p> | | |
| <p>Arbeitsaufwand:</p> <p>Gesamt: 150 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>60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>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 | | |
| <p>Modulteil: Safety-Critical Systems (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p> | | |

Inhalte:

Die Vorlesung betrachtet Methoden zur Einhaltung der funktionalen Sicherheit in Rechensystemen, sowie deren Entwurf und Analyse. 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. Verschiedene Methoden zur Fehlerinjektion, die zur Bewertung von fehlertolerierenden Systemen notwendig sind, werden kurz besprochen.

Literatur:

- D. Sorin: Fault Tolerant Computer Architecture, Morgan and Claypool, 2009
- S. Mukherjee: Architecture Design for Soft Errors, Morgan Kaufmann, 2008
- I. Koren, C.M. Krishna: Fault-Tolerant Systems, Morgan Kaufmann, 2007

Modulteil: Safety-Critical Systems (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Safety-Critical Systems

Klausur / Prüfungsdauer: 60 Minuten

| | | |
|---|---|---|
| Modul INF-0337: Seminar Embedded Systems (Master) <i>Seminar Embedded Systems (Master)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <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 Embedded Systems 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.</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 unter unterschiedlichen Gesichtspunkten</p> | | |
| <p>Arbeitsaufwand:</p> <p>Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p> | | |
| <p>Voraussetzungen:</p> <p>keine</p> | | |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Embedded Systems (Master) | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| <p>Inhalte:</p> <p>Im Seminar werden Themen aus dem Bereich Embedded Systems 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 Embedded Systems (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

Alle wichtigen Informationen zum Seminar finden Sie im Seminar-Wiki unter es-augsburg.de/seminar/.

Prüfung

Seminar Embedded Systems (Master)

Schriftlich-Mündliche Prüfung

| | | |
|--|---|---|
| Modul INF-0340: Projektmodul Embedded Systems <i>Project Module Embedded Systems</i> | | 10 ECTS/LP |
| Version 1.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet der Embedded Systems 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 |
| <p>Modulteil: Projektmodul Embedded Systems Lehrformen: Praktikum Sprache: Deutsch SWS: 1</p> |
| <p>Inhalte: Autonome Mitarbeit an aktuelle Forschungsthemen.</p> |
| <p>Literatur: wissenschaftliche Papiere, Handbücher</p> |
| <p>Zugeordnete Lehrveranstaltungen: Oberseminar Embedded Systems <i>*Veranstaltung wird online/digital abgehalten.*</i> Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren direkt einen Mitarbeiter, dessen Forschung Sie interessiert: www.es-augsburg.de Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte.</p> |

Prüfung

Projektmodul Embedded Systems

praktische Prüfung

| | | |
|--|---|---|
| Modul INF-0344: Seminar Software Engineering verteilter Systeme (MA) <i>Seminar Software Engineering of Distributed Systems (MA)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe20) 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. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p> | | |
| Voraussetzungen: Frühere Veranstaltung "Seminar über Software Engineering verteilter Systeme (MA)" (INF-0039) darf nicht belegt worden sein wegen Überschneidungen. | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Software Engineering verteilter Systeme (MA) | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| Inhalte: Aktuelle Software Engineering-Themen aus Industrie und Forschung. | | |
| Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt. | | |

Zugeordnete Lehrveranstaltungen:

Seminar Software Engineering verteilter Systeme (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

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

Seminar Software Engineering verteilter Systeme (MA)

Schriftlich-Mündliche Prüfung

| | | |
|--|---|---|
| Modul INF-0346: Seminar Automotive Software and Systems Engineering (MA) <i>Seminar Automotive Software and Systems Engineering (MA)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer | | |
| 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 & Systems 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. | | |
| 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 | | |
| Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium) | | |
| Voraussetzungen: Frühere Veranstaltung "Seminar Grundlagen des Software Engineering für Automotive Systems (MA)" (INF-0040) darf nicht belegt worden sein wegen Überschneidungen. | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Automotive Software and Systems Engineering (MA) Lehrformen: Seminar Sprache: Deutsch SWS: 2 | | |
| Inhalte: Aktuelle Software Engineering-Themen aus Industrie und Forschung. | | |

Literatur:

Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.

Zugeordnete Lehrveranstaltungen:

Seminar Automotive Software and Systems Engineering (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

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

Seminar Automotive Software and Systems Engineering (MA)

Schriftlich-Mündliche Prüfung

| | | |
|---|---|---|
| Modul INF-0348: Seminar Avionic Software and Systems Engineering (MA) <i>Seminar Avionic Software and Systems Engineering (MA)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer | | |
| 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 & Systems 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. | | |
| 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 | | |
| Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium) | | |
| Voraussetzungen: Frühere Veranstaltung "Seminar Grundlagen des Software Engineering für Avionic Systems (MA)" (INF-0041) darf nicht belegt worden sein wegen Überschneidungen. | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Avionic Software and Systems Engineering (MA) Lehrformen: Seminar Sprache: Deutsch SWS: 2 | | |
| Inhalte: Aktuelle Software Engineering-Themen aus Industrie und Forschung. | | |

Literatur:

Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt.

Prüfung

Seminar Avionic Software and Systems Engineering (MA)

Schriftlich-Mündliche Prüfung

| | | |
|---|---|---|
| Modul INF-0364: Seminar Software Engineering in sicherheitskritischen Systemen (MA) <i>Seminar Software Engineering in Safety- and Security-Critical Systems (MA)</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit WS20/21) 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 in sicherheitskritische Systemen und deren verwandten Disziplinen 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: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Software Engineering in sicherheitskritischen Systemen (MA) | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| Inhalte: Aktuelle Software Engineering-Themen aus Industrie und Forschung. | | |
| Literatur: Wird in der jeweiligen Kickoff-Veranstaltung vorgestellt. | | |
| Zugeordnete Lehrveranstaltungen: | | |

Seminar Software Engineering in sicherheitskritischen Systemen (Master) (Seminar)

Veranstaltung wird online/digital abgehalten.

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

Seminar Software Engineering in sicherheitskritischen Systemen (MA)

Schriftlich-Mündliche Prüfung

| | | |
|---|---|--|
| Modul INF-0367: Advanced Machine Learning and Computer Vision | | 5 ECTS/LP |
| Version 1.0.0 (seit WS20/21) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart | | |
| <p>Lernziele/Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul besitzen die Studierenden vertiefte fortgeschrittene Kenntnisse des maschinellen Lernens (Support Vector Machines und tiefe neuronale Netze und deren Grundbausteine) und des maschinellen Sehens (tiefe neuronale Netzarchitekturen und Systeme) und können diese anwenden. Sie sind in der Lage, wissenschaftlich komplexe Verfahren auf dem Gebiet der Bild-, Text-, Video- und Signalverarbeitung zu analysieren, zu verstehen und programmatisch umzusetzen, sowie die erlernten Prinzipien auf neue Fragestellungen geeignet anzuwenden. Sie entwickeln Fertigkeiten zum logischen, analytischen und konzeptionellen Denken im Bereich des maschinellen Lernens und Sehens.</p> <p>Schlüsselqualifikationen: fortgeschrittene mathematisch-formale Logik; kritisches Lesen und Analysieren von wissenschaftlichen Veröffentlichungen; 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. Vorlesung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 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)</p> | | |
| <p>Voraussetzungen: Kenntnisse in maschinellern Lernen und maschinellern Sehen (Master-Vorlesung INF-0092 "Multimedia II" bzw. INF-0316 "Machine Learning and Computer Vision")</p> | | <p>ECTS/LP-Bedingungen: Bestehen der Modulprüfung</p> |
| <p>Angebotshäufigkeit: jedes Wintersemester</p> | <p>Empfohlenes Fachsemester: ab dem 2.</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: Advanced Machine Learning and Computer Vision (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p> | | |
| <p>Inhalte: Die Vorlesung gibt einen vertieften Einblick in alle Aspekte des maschinellen Lernens und des maschinellen Sehens. Die erlernten Konzepte werden in den Übungen anhand von erfolgreichen Beispielen aus der Praxis ausprobiert, geübt, analysiert und bewertet. Die Inhalte der Vorlesung umfassen: Support Vector Machines, Grundbausteine von tiefen Neuronalen Netzen (Layerstrukturen, Normalisierung, Attention-Mechanismen), sowie aktuelle Referenzarchitekturen und -system für Bild-, Text-, Videoverarbeitung und deren Kombination mit weiteren Sensorsignalen.</p> | | |
| <p>Literatur: Literaturhinweise werden zu Beginn des Semesters bekanntgegeben.</p> | | |
| <p>Zugeordnete Lehrveranstaltungen:</p> | | |

Advanced Machine Learning and Computer Vision (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Modulteil: Advanced Machine Learning and Computer Vision (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Advanced Machine Learning and Computer Vision (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Advanced Machine Learning and Computer Vision (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Prüfung kann jedes Semester in der Prüfungszeit abgelegt werden.

| | | |
|--|---|---|
| Modul INF-0368: Embedded Hardware <i>Embedded Hardware</i> | | 5 ECTS/LP |
| Version 1.0.0 (seit WS20/21) Modulverantwortliche/r: Prof. Dr. Sebastian Altmeyer | | |
| <p>Lernziele/Kompetenzen:</p> <p>Die Vorlesung vermittelt grundlegende und weiterführende Kenntnisse zu Hardwarekomponenten (z.B. GPIO, Hardware-Timer, ...) wie sie in nahezu allen eingebetteten Systemen vorkommen. Außerdem werden die Funktionsprinzipien von verschiedenen Sensoren und Aktuatoren vorgestellt. Ein besonderer Fokus liegt dabei auf deren Einsatz zur Interaktion mit der physikalischen Welt und auf Besonderheiten bei deren Anwendung. Die theoretischen Grundlagen werden sich am aktuellen Forschungsstand orientieren und eine weitere Beschäftigung der Studierenden mit dem Thema eingebetteter Hardware auf wissenschaftlichen Niveau ermöglichen.</p> <p>Die Vorlesung wird den Studierenden die Fähigkeit vermitteln verschiedene eingebettete Systeme anhand ihrer Hardwareanforderungen zu unterscheiden und einzuordnen. Die Studierenden lernen aktuelle Methoden zur Programmierung der Komponenten anzuwenden, zu vergleichen, und im Hinblick auf eine effiziente Nutzung kritisch zu analysieren. Dies beinhaltet die Auswahl und Beurteilung von für das System relevanten Hardwarekomponenten, sowie die Entwicklung einer für ein bestehendes Problem optimierten Konfiguration der Komponenten. Die Vorlesung behandelt zudem verschiedene Sensoren und Aktuatoren, und wird näher auf die Umwandlung zwischen einer analogen, physikalischen Größe und einer digitalen, elektrischen Größe eingehen. Die Studierenden werden hierbei in die Lage versetzt, physikalische Größen anhand ihrer Eigenschaften zu klassifizieren und eine Entscheidung über die für die Umwandlung benötigten Sensoren, Aktuatoren und Hardwarekomponenten zu treffen.</p> <p>Der Lehrstoff wird anhand von Fallstudien z.B. aus den Bereich Automobil exemplifiziert und von den Studierenden anhand eines einfachen eingebetteten Systems angewendet.</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:</p> <p>Gesamt: 150 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>60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>30 Std. Übung (Präsenzstudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| <p>Modulteil: Embedded Hardware (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch / Englisch</p> <p>SWS: 2</p> | | |

Inhalte:

- Modellierung von Sensoren und Aktuatoren
- Grundlagen in Elektronik
- Hardwarekomponenten von eingebetteten Prozessoren und SoC
- Serielle Schnittstellen
- Sensoren und Eingabegeräte
- Aktuatoren und Ausgabegeräte

Literatur:

- Yifeng Zhu, Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C, E-Man Press LLC, 2017
- Edward Ashford Lee, Sanjit Arunkumar Seshia, Introduction to Embedded Systems - A Cyber-Physical Systems Approach, MIT Press, 2017
- Elecia White, Making Embedded Systems, O'Reilly Media, 2012
- John Catsoulis, Designing Embedded Hardware, O'Reilly Media, 2005
- Rüdiger R. Asche, Embedded Controller: Grundlagen und praktische Umsetzung für industrielle Anwendungen, Springer, 2016
- Hermann Winner, Stephan Hakuli, Felix Lotz, Christina Singer Hrsg., Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort, Springer, 2015
- Bernhard Grimm, Gregor Häberle, Heinz Häberle, u.a., Fachkunde Industrieelektronik und Informationstechnik, Europa Lehrmittel, 2003

Zugeordnete Lehrveranstaltungen:

Embedded Hardware (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Eingebettete Systeme sind Computer Systeme, deren vorrangige Aufgabe es ist die umgebende physikalische und technische Welt zu erfassen bzw. zu beeinflussen. Diese kann beispielsweise durch regelmäßiges Messen physikalischer Größen oder durch die Steuerung bzw. Regelung von technischen Anlagen geschehen. Um diese Aufgaben zu bewerkstelligen benötigen eingebettete Prozessoren diverse Hardwarekomponenten, die es ihnen ermöglicht mit der Außenwelt zu interagieren. Zusätzlich werden Sensoren und Aktuatoren benötigt. Sie übernehmen die Umwandlung von physikalischen Größen in elektrische Signale und umgekehrt. In dieser Vorlesung beschäftigen wir uns sowohl mit den beschriebenen Hardwarekomponenten als auch mit verschiedenen Sensoren und Aktuatoren. Wir lernen wie ein Prozessor mit Hilfe dieser Komponenten mit seiner Umgebung interagieren kann und welchen Einfluss die Komponenten auf die Wandlung von physikalischen Größen zu einem digitalen Wert (und umgekehrt) haben kann.
 ... (weiter siehe Digicampus)

Modulteil: Embedded Hardware (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Embedded Hardware (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Embedded Hardware

Klausur / Prüfungsdauer: 60 Minuten

| | | |
|---|--|--|
| Modul INF-0236: Digitale Regelsysteme <i>Digital Control Systems</i> | | 6 ECTS/LP |
| Version 1.1.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Christoph Ament | | |
| <p>Lernziele/Kompetenzen: Kostengünstige Sensoren und ebenfalls sehr preiswert verfügbare Rechenleistung erlauben es heute, Prozesse umfassend zu erfassen bzw. aufwendige Algorithmen zur Signalverarbeitung einzusetzen. Im Zusammenspiel mit dem physikalischen System lässt sich so ein "smartes" Gesamtsystem erreichen. Doch wie kann man die großen Freiheiten im Entwurf des IT-Systems sinnvoll und zielführend nutzen?</p> <p>Die Vorlesung vermittelt Ihnen Werkzeuge, für den Entwurf dieses "digitalen Regelsystems". Als Grundlage lernen Sie, zeitdiskrete dynamische Systeme zu beschreiben und zu analysieren. Sie können Verfahren zur zeitlichen Diskretisierung anwenden. Es werden Konzepte und Module für den Aufbau eines digitalen Regelsystems vorgestellt. Sie können diese einordnen und auf eine Projektaufgabe übertragen. Dazu können Sie geeignete modellbasierte Entwurfsverfahren anwenden, um eine entsprechende Software zu entwickeln und Ihre Diagnose- oder Regelungsaufgabe zu lösen.</p> | | |
| <p>Arbeitsaufwand: Gesamt: 180 Std. 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium)</p> | | |
| <p>Voraussetzungen: Als Voraussetzung für diese Veranstaltung werden grundlegende Kenntnisse der Regelungstechnik (z.B. "Mess- und Regelungstechnik") sowie der Systemdarstellung im Zustandsraum (z.B. "Modellbildung, Identifikation und Simulation dynamischer Systeme" oder "Regelungstechnik 2") aus dem Bachelor-Studium empfohlen. Diese Veranstaltung "Digitale Regelsysteme" wird als Basisveranstaltung im Master Ingenieurinformatik empfohlen.</p> | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 5 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| <p>Modulteil: Digitale Regelsysteme (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 3</p> | | |

Inhalte:

Controller, die Systeme und Prozesse überwachen, steuern und regeln, werden heute zumeist als Software auf einem Digitalrechner implementiert. In dieser Veranstaltung werden Methoden vermittelt, mit denen Sie diese Algorithmen systematisch und modellbasiert auch für komplexe Systeme entwerfen können.

Digitalrechner arbeiten in diskreten Zeitschritten. Daher ist es effizient, eine zeitdiskrete Systemdarstellung zu Grunde zu legen. In Teil A der Vorlesung wird die Ihnen bekannte zeitkontinuierliche Systembeschreibung (z.B. durch eine Übertragungsfunktion $G(s)$) auf eine zeitdiskrete Darstellung erweitert und die Analyse von Stabilität, Steuer- und Beobachtbarkeit entsprechend eingeführt.

Teil B widmet sich dem Entwurf von Reglern. Dazu wird zunächst unter dem Begriff "Rapid Control Prototyping" eine durchgängige Vorgehensweise entwickelt. Es wird gezeigt, wie kontinuierlich entworfene Regler diskretisiert werden können und welche Vorteile demgegenüber der direkte zeitdiskrete Entwurf besitzt. Für den zeitdiskreten Entwurf werden ausgewählte Reglerentwurfsverfahren für lineare und nichtlineare Systeme vorgestellt.

Neben der Regelung gewinnen Aufgaben der Prozessüberwachung, Diagnose und Adaption zunehmend an Bedeutung, die ebenfalls als Teil eines Regelsystems in Software realisiert werden können. Dies wird im Teil C der Vorlesung gezeigt. Ausgehend von einer stochastischen Modellerweiterung werden Algorithmen zur Parameterschätzung vorgestellt, die zur Diagnose und Adaption genutzt werden können. Daraus wird schließlich die Schätzung dynamischer Zustände (mittels Kalman-Filter) entwickelt und auf nichtlineare Systeme erweitert.

Gliederung:

1. Einführung: Ziele und Aufbau eines digitalen Regelsystems

Teil A: Zeitdiskrete Systeme

2. Darstellung im Zeitbereich

3. Darstellung im Bildbereich (z-Transformation)

4. Analyse von Systemeigenschaften

Teil B: Modellbasierter Reglerentwurf

5. Rapid Control Prototyping

6. Reglerentwurfsverfahren (Eigenwertvorgabe, Entwurf auf Endliche Einstellzeit, Optimalregler, Modell Predictive Control)

Teil C: Modellbasierte Diagnose

7. Grundlagen stochastischer Systeme

8. Schätzung von Parametern

9. Schätzung von Zuständen (Kalman-Filter)

10. Erweiterung auf nichtlineare Systeme

11. Technische Diagnose

Literatur:

Literatur (Vorlesung):

Grundlagen und Wiederholung:

- Föllinger, O: Regelungstechnik – Einführung in die Methoden und ihre Anwendung, VDE Verlag, 12. Auflage, 2016.
- Lunze, J.: Regelungstechnik 1 – Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen, Springer, 9. Auflage, 2013.

Zur Vorlesung:

- Lunze, J.: Regelungstechnik 2 – Mehrgrößensystem, Digitale Regelung, Springer, 7. Auflage, 2013.
- Lunze, J.: Automatisierungstechnik – Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme, Oldenbourg, 3. Auflage 2012.
- Abel, D, Bollig, A.: Rapid Control Prototyping: Methoden und Anwendungen, Springer, 2006.

Zugeordnete Lehrveranstaltungen:

Digitale Regelsysteme (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Modulteil: Digitale Regelsysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu digitale Regelsysteme (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Digitale Regelsysteme

Schriftlich-Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

Die Wiederholungsprüfung findet nach der Vorlesungszeit des Sommersemesters statt.

| | | |
|--|---|---|
| Modul INF-0237: Projektmodul Regelungstechnik <i>Project Module on Control Engineering</i> | | 10 ECTS/LP |
| Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Christoph Ament | | |
| <p>Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet der Regelungstechnik 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; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten</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 |
| <p>Modulteil: Projektmodul Regelungstechnik Lehrformen: Praktikum Sprache: Deutsch SWS: 1</p> |
| <p>Inhalte: Mitarbeit an aktuellen Forschungsthemen des Lehrstuhls</p> |
| <p>Literatur: abhängig von dem konkreten Projekt: wissenschaftliche Aufsätze, Dokumentation</p> |
| <p>Zugeordnete Lehrveranstaltungen: Oberseminar zur Regelungstechnik <i>*Veranstaltung wird online/digital abgehalten.*</i> Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren direkt einen</p> |

Mitarbeiter, dessen Forschung Sie interessiert: <https://www.informatik.uni-augsburg.de/lehrstuehle/rt/> Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Prüfung

Projektabschluss

Praktikum

| | | |
|---|---|---|
| Modul INF-0238: Digitale Fabrik <i>Digital Factory</i> | | 6 ECTS/LP |
| Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr.-Ing. Johannes Schilp | | |
| <p>Lernziele/Kompetenzen: Ziel der Vorlesung Digitale Fabrik ist es, den Studierenden die Konzepte der Digitalen Fabrik und ein vertieftes Verständnis für Produktionsorganisation und -abläufe zu vermitteln. Sie können Anwendungsmöglichkeiten der Digitalen Fabrik im Bereich der Planung und Simulation darstellen. Die Studenten sind darüber hinaus fähig die Einsatzmöglichkeiten von Werkzeugen der Digitalen Fabrik in produzierenden Unternehmen zu analysieren und können die Potentiale im Kontext konkreter Fragestellungen bewerten. Sie lösen einfache Simulationsaufgaben mithilfe einer verbreiteten Simulationssoftware und entwerfen darauf aufbauend selbstständig ein anspruchsvolleres Modell.</p> <p>Schlüsselqualifikation: Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Abstraktionsfähigkeit, anwendungsorientierte Problemlösung, Abwägen von Lösungsansätzen, überzeugende Darstellung und Dokumentation von Konzepten und Ergebnissen</p> | | |
| <p>Arbeitsaufwand: Gesamt: 180 Std. 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Übung (Präsenzstudium) 45 Std. Vorlesung (Präsenzstudium)</p> | | |
| <p>Voraussetzungen: Empfohlen wird, dass Sie eines der folgenden Module vorher belegt haben:</p> <ul style="list-style-type: none"> • INF-0196: Produktionsinformatik • INF-0197: Prozessmodellierung und Produktionssteuerung • INF-0260: Produktionstechnik | | |
| <p>Angebotshäufigkeit: jedes Wintersemester</p> | <p>Empfohlenes Fachsemester: 1.</p> | <p>Minimale Dauer des Moduls: 1 Semester</p> |
| <p>SWS: 5</p> | <p>Wiederholbarkeit: siehe PO des Studiengangs</p> | |
| Modulteile | | |
| <p>Modulteil: Digitale Fabrik (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 3</p> | | |

Inhalte:

Nach VDI 4499 versteht man unter Digitaler Fabrik „ein Netzwerk von digitalen Modellen, Methoden und Werkzeugen – u.a. der Simulation und 3D-Visualisierung“ sowie deren Einbindung in das unternehmensweite Datenmanagement.

Folgende Themenbereiche werden in der Vorlesung behandelt:

- Einsatzbereiche der Werkzeuge und Technologien zur Fabrikplanung und -gestaltung
- Einsatzbereiche der Werkzeuge und Technologien im Produktionsumfeld: digitale Unterstützung in Fertigung und Montage sowie Optimierung von Strukturen, Prozessen und Ressourcen in der Fabrik
- Potentiale, Nutzen und Vorteile für Unternehmen
- Modellierungs- und Simulationsansätze
- Augmented und Virtual Reality
- Überblick über verbreitete Software
- Praxisbeispiele

Literatur:

wird in der Vorlesung bekannt gegeben

Zugeordnete Lehrveranstaltungen:

Digitale Fabrik (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Modulteil: Digitale Fabrik (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Inhalte:

Wiederholung und Vertiefung der Lehrinhalte aus der Vorlesung mithilfe von Übungen und Praxisbeispielen

Zugeordnete Lehrveranstaltungen:

Übung zu Digitale Fabrik (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung

Digitale Fabrik

Klausur / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Wiederholungsprüfung findet nach der Vorlesungszeit des Sommersemesters statt.

| | | |
|---|---|---|
| Modul INF-0244: Projektmodul Produktionsinformatik <i>Project Module on Digital Manufacturing</i> | | 10 ECTS/LP |
| Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr.-Ing. Johannes Schilp | | |
| Inhalte: Mitarbeit an aktuellen Forschungsthemen des Lehrstuhls | | |
| Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet Produktionsinformatik 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. 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; Bewertung von Lösungsansätzen, Verfahren, Techniken und Technologien unter unterschiedlichen Gesichtspunkten | | |
| Bemerkung: Ersetzt "Projektmodul Industrie 4.0" (INF-0239) | | |
| Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium) | | |
| 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 Produktionsinformatik Lehrformen: Praktikum Sprache: Deutsch SWS: 1 |
| Inhalte: Mitarbeit an aktuellen Forschungsthemen des Lehrstuhls |
| Literatur: Abhängig von dem konkreten Projekt: wissenschaftliche Papiere, Dokumentationen |
| Zugeordnete Lehrveranstaltungen: |

Oberseminar zur Produktionsinformatik

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren direkt einen Mitarbeiter, dessen Forschung Sie interessiert: <https://www.uni-augsburg.de/de/fakultaet/fai/informatik/prof/pi/lehre/fm-pm-seminar/> Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Prüfung

Projektabnahme

Praktikum

| | | |
|---|---|---|
| Modul INF-0245: Intelligent vernetzte Produktion <i>Intelligently Networked Manufacturing</i> | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr.-Ing. Johannes Schilp | | |
| <p>Lernziele/Kompetenzen: Ziel der Vorlesung Intelligent vernetzte Produktion ist es, den Studierenden ein vertieftes Verständnis über Vernetzung im Kontext produzierender Unternehmen zu vermitteln. Sie sind in der Lage, Produktionsnetzwerke zu beschreiben und Vernetzungen auf Mikro- und Makroebene zu analysieren. Die Studierenden können resultierende Optimierungsmöglichkeiten darstellen und reflektiert bewerten. Sie sind fähig, erlernte Methoden zur Optimierung im Umfeld industrieller Produktion anzuwenden.</p> <p>Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Abstraktionsfähigkeit, anwendungsorientierte Problemlösung, Abwägen von Lösungsansätzen, überzeugende Darstellung und Dokumentation von Konzepten und Ergebnissen</p> | | |
| <p>Arbeitsaufwand: Gesamt: 150 Std. 60 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) 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium)</p> | | |
| <p>Voraussetzungen: Empfohlen wird, dass Sie eines der folgenden Module vorher belegt haben:</p> <ul style="list-style-type: none"> • INF-0196: Produktionsinformatik • INF-0197: Prozessmodellierung und Produktionssteuerung • INF-0260: Produktionstechnik | | |
| Angebotshäufigkeit: jedes Sommersemester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| <p>Modulteil: Intelligent vernetzte Produktion (Vorlesung) Lehrformen: Vorlesung Dozenten: Prof. Dr.-Ing. Johannes Schilp Sprache: Deutsch SWS: 2</p> | | |

Inhalte:

Im Zuge der Vorlesung Intelligent vernetzte Produktion werden den Studierenden folgende Ebenen der Vernetzung in der industriellen Produktion vermittelt:

- Mikroebene (Werk Fokus): Cyber-physische Vernetzung in der Produktion
- Makroebene (Netzwerk Fokus): Aufbau und Betrieb globaler unternehmensinterner und unternehmensübergreifender Produktionsnetzwerke sowie Grundlagen des Supply Chain Managements
- Industriebetriebe als wichtiger Bestandteil intelligenter Stromnetze
- Technologien sowie mögliche Ausprägungen und Strategien zur Vernetzung in den jeweiligen Bereichen werden besprochen.

Resultierende Optimierungsmöglichkeiten durch Abgleich von realer und digitaler Welt werden aufgezeigt. Relevante Praxisbeispiele aus dem Bereich der vernetzten Produktion werden ebenso erörtert wie aktuelle Forschungsprojekte.

Modulteil: Intelligent vernetzte Produktion (Übung)

Lehrformen: Übung

Dozenten: Prof. Dr.-Ing. Johannes Schilp

Sprache: Deutsch

SWS: 2

Inhalte:

Wiederholung und Vertiefung der Lehrinhalte aus der Vorlesung mithilfe von Übungen, Praxisbeispielen und Fallstudien.

Prüfung

Intelligent vernetzte Produktion

Schriftlich-Mündliche Prüfung / Prüfungsdauer: 90 Minuten

Beschreibung:

Die Wiederholungsprüfung findet nach der Vorlesungszeit des Wintersemesters statt.

| | | |
|---|---|---|
| Modul INF-0246: Seminar Industrie 4.0 <i>Seminar on Industry 4.0</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr.-Ing. Johannes Schilp | | |
| <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 Industrie 4.0 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>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 Semester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| <p>Modulteil: Seminar Industrie 4.0 Lehrformen: Seminar Sprache: Deutsch SWS: 2 ECTS/LP: 4.0</p> | | |
| Inhalte: Mitarbeit an aktuellen Forschungsthemen des Lehrstuhls. | | |
| Literatur: abhängig vom jeweiligen Thema | | |
| Zugeordnete Lehrveranstaltungen: | | |

Seminar zu Industrie 4.0 (Seminar)

Veranstaltung wird online/digital abgehalten.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

| | | |
|---|---|---|
| Modul INF-0247: Praktikum für Produktionsinformatik (Vertiefung) <i>Practical Module on Digital Manufacturing</i> | | 6 ECTS/LP |
| Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr.-Ing. Johannes Schilp | | |
| <p>Lernziele/Kompetenzen: Die Studierenden durchlaufen in Kleingruppen anhand eines industrienahen Beispiels den digitalen Produktentwicklungsprozess. Sie sind in der Lage selbstständig ingenieurtechnische Aufgaben zu analysieren und Lösungskonzepte zu erarbeiten. Das Wissen aus den ingenieurwissenschaftlichen Grundlagen wird durch Anwendungsbeispiele vertieft. Sie sind in der Lage ausgewählte CAx-Programme (CATIA V5, PlantSim, FreeCAD) für die Entwicklung eines Produkts anzuwenden. Das Praktikum bietet eine Einführung in folgende Themenbereiche:</p> <ol style="list-style-type: none"> 1. CAD-Konstruktion 2. FEM-Analyse 3. Topologieoptimierung 4. Produktionsplanung 5. Mathematische Optimierung <p>Schlüsselqualifikationen: Team- und Kommunikationsfähigkeit, strukturiertes und gewissenhaftes Arbeiten, anwendungsorientierte Problemlösung, Ergebnisbewertung und -dokumentation, Abwägen von Lösungsansätzen, Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Abstraktionsfähigkeit</p> | | |
| <p>Arbeitsaufwand: Gesamt: 180 Std. 10 Std. Vorbereitung von Präsentationen (Selbststudium) 40 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 10 Std. Vorlesung (Präsenzstudium) 60 Std. Praktikum (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> | | |
| Voraussetzungen: Grundkenntnisse in CATIA V5 empfehlenswert | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| <p>Modulteil: Praktikum für Produktionsinformatik (Vertiefung) Lehrformen: Praktikum Sprache: Deutsch SWS: 4 ECTS/LP: 6.0</p> | | |
| <p>Inhalte: Die Studenten bearbeiten in Kleingruppen anwendungsorientierte Aufgaben zu Themenbereichen im industriellen Umfeld. Der zeitliche Ablauf dieses Praktikums wird in Digicampus bekannt gegeben.</p> | | |
| <p>Zugeordnete Lehrveranstaltungen: Praktikum für Produktionsinformatik (Vertiefung) (Praktikum) <i>*Veranstaltung wird als Hybrid/gemischt abgehalten.*</i> Es ist geplant, das Praktikum als BLOCKVERANSTALTUNG vom 08.03.2021-26.03.2021 stattfinden zu lassen.</p> | | |

Prüfung

Praktikum für Produktionsinformatik (Vertiefung)

Praktikum

| | | |
|--|---|---|
| Modul INF-0249: Seminar zu speziellen Kapiteln der Regelungstechnik <i>Seminar on Special Topics in Control Engineering</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Christoph Ament | | |
| 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 Regelungstechnik 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; | | |
| Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) | | |
| Voraussetzungen: Grundkenntnisse auf dem Gebiet der Regelungstechnik z.B. aus dem Bachelor-Studium. | | |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar zu speziellen Kapiteln der Regelungstechnik Lehrformen: Seminar Sprache: Deutsch SWS: 2 ECTS/LP: 4.0 | | |

Inhalte:

Können Sie auf Basis Ihrer bisher im Studium erworbenen Kenntnisse aktuelle Veröffentlichungen auf dem Gebiet der Ingenierinformatik oder den Ingenieurwissenschaften erschließen und einordnen? Das gehen wir im Seminar an!

Wir widmen uns einem Schwerpunktthema der System- und Regelungstechnik, das Sie jeweils zu Beginn des Semesters der Webseite des Lehrstuhls entnehmen können. Wir verschaffen uns einen Überblick über aktuelle Veröffentlichungen. Ihre Aufgabe ist es, einen ausgewählten Beitrag zu bearbeiten und in einem kurzen Vortrag vorzustellen.

Literatur:

abhängig vom jeweiligen Thema

Zugeordnete Lehrveranstaltungen:

Seminar zu speziellen Kapiteln der Regelungstechnik (Seminar)

Veranstaltung wird als Hybrid/gemischt abgehalten.

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

| | | |
|--|--|--|
| Modul INF-0264: Nichtlineare Regelsysteme <i>Nonlinear Control Systems</i> | | 5 ECTS/LP |
| Version 1.1.0 (seit WS17/18) Modulverantwortliche/r: Prof. Dr. Christoph Ament | | |
| Lernziele/Kompetenzen: Die Hörer und Hörerinnen der Vorlesung kennen die Eigenschaften nichtlinearer Systeme und können diese von linearen Systemen unterscheiden. Sie sind in der Lage, die Eigenschaften nichtlinearer Systeme (z.B. in der Zustandsebene) zu analysieren. Sie können Konzepte zur Stabilitätsanalyse nach Lyapunov erklären und anwenden. Es werden ausgewählte Verfahren zum Entwurf nichtlinearer Regelungen eingeführt. Die Hörer und Hörerinnen können diese Konzepte erklären und auf nichtlineare Systeme anwenden. | | |
| Arbeitsaufwand: Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 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) | | |
| Voraussetzungen: Als Voraussetzung für diese Veranstaltung werden grundlegende Kenntnisse der Regelungstechnik (z.B. "Mess- und Regelungstechnik") sowie der Systemdarstellung im Zustandsraum (z.B. "Modellbildung, Identifikation und Simulation dynamischer Systeme" oder "Regelungstechnik 2") aus dem Bachelor-Studium empfohlen. | | ECTS/LP-Bedingungen: Bestehen der Modulprüfung |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Nichtlineare Regelsysteme (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2 | | |

Inhalte:

Bei genauerer Betrachtung weisen praktisch alle Systeme nichtlineare Eigenschaften aus. Dies können physikalische Effekte der Strecke wie Reibung oder Lose sein. Aber auch durch eine Regelung werden Nichtlinearitäten eingebracht, z.B. durch Schaltvorgänge oder Stellgrößenbeschränkungen. Möchte man der Dynamik von Systemen gerecht werden, um z.B. optimale Leistungsfähigkeit von Robotern zu erreichen, muss man diese nichtlinearen Effekte charakterisieren und beherrschen können. In dieser Vorlesung stehen die Charakterisierung als auch die Regelung nichtlinearer Systeme im Mittelpunkt.

Anders als bei linearen Systemen können Eigenschaften wie Stabilität nicht global dem System als ganzem zugeordnet werden, sondern müssen spezifischen Ruhelagen zugeordnet werden. Dies wird in der Vorlesung zunächst an Beispielen diskutiert. Nach der Vermittlung von mathematischen Grundlagen werden zweidimensionale Systeme in der Zustandsebene betrachtet.

Die Lyapunov-Theorie erlaubt die Analyse von Stabilität nichtlinearer Systeme und kann auch für den Reglerentwurf genutzt werden. In analoger Weise wird die Passivität zur Analyse von Systemen eingeführt und dann auch für den Entwurf von Regelungen genutzt.

Schließlich wird die exakte Linearisierung für den Reglerentwurf vorgestellt.

Gliederung:

1. Einführung: Beispiele und Eigenschaften nichtlinearer Systeme
2. Grundbegriffe
3. Zustandsebene
4. Stabilität von Ruhelagen (Lyapunov-Theorie)
5. Reglerentwurf mit Hilfe der Lyapunov-Theorie
6. Passivität
7. Passivitätsbasierter Reglerentwurf (Backstepping)
8. Entwurf durch exakte Linearisierung

Literatur:

- J. Adamy: Nichtlineare Systeme und Regelungen, 2. Auflage, Springer Verlag, 2014.
- H. K. Khalil: Nonlinear Control, Global Edition (Englisch), Pearson, 2014.

Zugeordnete Lehrveranstaltungen:**Nichtlineare Regelsysteme** (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Modulteil: Nichtlineare Regelsysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:**Übung zu nichtlinearen Regelsystemen** (Übung)

Veranstaltung wird online/digital abgehalten.

Prüfung**Nichtlineare Regelsysteme**

Schriftlich-Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

Die Wiederholungsprüfung findet nach der Vorlesungszeit des Sommersemesters statt

| | | |
|--|---|---|
| Modul INF-0272: Intelligente Signalanalyse in der Medizin <i>Intelligent Signal Analysis in Medicine</i> | | 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> | | |
| Voraussetzungen: Knowledge of basic mathematic lectures should be present. | | |
| 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: Intelligente Signalanalyse in der Medizin (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Englisch</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

SWS: 2

Prüfung

Intelligente Signalanalyse in der Medizin (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0273: Praktikum Mobile Sensing for Fitness and Well-being <i>Practical Module Sensing for Fitness and Wellbeing</i> | | 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: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| <p>Modulteil: Praktikum Mobile Sensing for Fitness and Wellbeing</p> <p>Lehrformen: Praktikum Sprache: Englisch Angebotshäufigkeit: jedes Wintersemester SWS: 4</p> | | |
| <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> | | |

Literatur:

Wird vom Dozenten / von der Dozentin bekannt gegeben

Prüfung

Praktikum Mobile Sensing for Fitness and Wellbeing

Praktikum

| | | |
|---|---|---|
| Modul INF-0301: Objektorientierte Methoden der Modellbildung und Simulation <i>Object Oriented Modeling and Simulation</i> | | 6 ECTS/LP |
| Version 1.2.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Lars Mikelsons | | |
| Lernziele/Kompetenzen: Fachbezogene Kompetenzen: Nach der erfolgreichen Teilnahme an diesem Modul können die Studierenden fortgeschrittene Methoden der Modellbildung und Simulation anwenden, um virtuelle Prototypen (=mathematische Modelle) domänenübergreifender physikalischer Systeme aufzustellen. Sie können die Modellierung komplexer Systeme durch Konstruktion eigener und Anwendung vorhandener Modellbibliotheken realisieren. Methodische Kompetenzen: Die Studierenden sind in der Lage die unterschiedlichen Aspekte der Methodik der objektorientierten Modellbildung und der signalbasierten Modellbildung zu bewerten. Sie können darüber hinaus die Folgen von Modellierungsansätzen auf die symbolische Gleichungsverarbeitung und Numerik abschätzen und somit Modellqualität analysieren. Sie können basierend auf einer Beschreibung des physikalischen Systems ein objektorientiertes mathematisches Modell aufstellen. Fachübergreifende Kompetenzen: Die Studierenden sind in der Lage gewisse ingenieurwissenschaftliche Fragestellungen mit Hilfe von Informationstechnologie zu lösen. Sie können Abstraktionen physikalischer Systeme erzeugen und diese in informationstechnologische Zusammenhänge einbinden. Schlüsselkompetenzen: Fortgeschrittene Fertigkeit ingenieurwissenschaftliche Fragestellungen, insbesondere aus dem Bereich des Entwurfs mechatronischer Systeme, mit Hilfe von virtuellen Prototypen zu beantworten; Aufbau wiederverwendbarer Modellbibliotheken; Beurteilung und Analyse von mathematischen Modellen. | | |
| Arbeitsaufwand: Gesamt: 180 Std. 45 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 45 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 5 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Objektorientierte Methoden der Modellbildung und Simulation (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 3 | | |

Inhalte:

In der Vorlesung werden moderne Methoden für die Simulation technischer Systeme behandelt. Hierfür wird die Methode der objektorientierten Modellierung vorgestellt, welche es erlaubt verschiedene physikalische Domänen mit dem demselben Ansatz zu modellieren und somit auch zu kombinieren. Darüber hinaus werden Algorithmen sowohl für die Verarbeitung als auch für die Lösung entsprechender Modelle präsentiert.

Die Studierenden lernen Modelle mit der Modellierungssprache Modelica zu erstellen sowie fremde Modelle zu verstehen und zu nutzen.

Literatur:

"Principles of Object-Oriented Modeling and Simulation with Modelica 3.3" von Peter Fritzson

"Continuous System Simulation" von Francois Cellier

"Modelica by Example" von Michael Tiller Freies online Buch

<http://book.xogeny.com/>

Zugeordnete Lehrveranstaltungen:

Objektorientierte Methoden der Modellbildung und Simulation (Vorlesung)

Veranstaltung wird online/digital abgehalten.

Im Zuge der Vernetzung und Automatisierung von technischen Systemen werden zunehmend reale Prototypen durch virtuelle Prototypen ersetzt. In dieser Veranstaltung lernen die Studierenden moderne Methoden für die Erstellung virtueller Prototypen. Inhaltsübersicht • Vorstellung der Modellierungssprache Modelica • Erstellung von eigenen Modellen am Beispiel der ebenen Mechanik • Erstellung von effizienten mathematischen Modellen aus objektorientierten Modellen • Algorithmen für Lösung von mathematischen Modellen

Modulteil: Objektorientierte Methoden der Modellbildung und Simulation (Übung)

Lehrformen: Übung

Dozenten: Prof. Dr. Lars Mikelsons

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Objektorientierte Methoden der Modellbildung und Simulation (Übung)

Veranstaltung wird online/digital abgehalten.

Aufgrund der eingesetzten Softwarelizenzen muss die Anmeldung zur Veranstaltung gesperrt werden. Alle Teilnehmer der Vorlesung "Objektorientierte Methoden der Modellbildung und Simulation" werden automatisch zu Semesterbeginn auch in die Übung eingetragen.

Prüfung

Objektorientierte Methoden der Modellbildung und Simulation (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

Die Wiederholungsprüfung findet zu Beginn des folgenden Semesters statt.

| | | |
|---|---|---|
| Modul INF-0302: Projektmodul Mechatronik <i>Project Module on Mechatronics</i> | | 10 ECTS/LP |
| Version 1.0.0 (seit WS18/19) Modulverantwortliche/r: Prof. Dr. Lars Mikelsons | | |
| Lernziele/Kompetenzen: Fachbezogene Kompetenzen: Nach der Teilnahme am Projektmodul verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet der Mechatronik 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. Schlüsselkompetenzen: 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; | | |
| 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 | |
| Modulteile | | |
| Modulteil: Projektmodul Mechatronik Lehrformen: Praktikum Dozenten: Prof. Dr. Lars Mikelsons Sprache: Deutsch / Englisch SWS: 1 | | |
| Inhalte: Im Seminar werden aktuelle Forschungsthemen aus dem Bereich der Mechatronik 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, ein prototypischer Aufbau oder eine prototypische Implementierung sowie ein Vortrag über das behandelte Thema dar. | | |
| Literatur: abhängig von dem konkreten Projekt: wissenschaftliche Aufsätze, Dokumentation | | |
| Zugeordnete Lehrveranstaltungen: | | |

Oberseminar zur Mechatronik

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn du ein solches Modul am Lehrstuhl für Mechatronik belegen willst, komm doch einfach auf uns zu (persönlich, Mail, zoom, ...). Wir freuen uns mit dir unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten zu diskutieren. Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Prüfung

Projektabnahme

Praktikum

| | | |
|---|---|---|
| Modul INF-0304: Seminar zu aktuellen Forschungsthemen der Mechatronik <i>Seminar Current Research Topics in Mechatronics</i> | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe19) Modulverantwortliche/r: Prof. Dr. Lars Mikelsons | | |
| Lernziele/Kompetenzen: Fachbezogene Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet der Mechatronik 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; | | |
| Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium) | | |
| Voraussetzungen: Grundkenntnisse auf dem Gebiet der Mechatronik z.B. aus dem Bachelor-Studium. | | |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |

| |
|--|
| <p>Modulteile</p> |
| <p>Modulteil: Seminar zu aktuellen Forschungsthemen der Mechatronik Lehrformen: Seminar Dozenten: Prof. Dr. Lars Mikelsons Sprache: Deutsch / Englisch SWS: 2 ECTS/LP: 4.0</p> |
| <p>Inhalte:</p> <p>Können Sie auf Basis Ihrer bisher im Studium erworbenen Kenntnisse aktuelle Veröffentlichungen auf dem Gebiet der Ingenieurinformatik oder der Ingenieurwissenschaften erschließen oder einordnen? Das gehen wir im Seminar an!</p> <p>Wir widmen uns einem Schwerpunktthema der Mechatronik, das Sie jeweils zu Beginn des Semesters der Website des Lehrstuhls entnehmen können. Wir verschaffen uns einen Überblick über aktuelle Veröffentlichungen. Ihre Aufgabe ist es, einen ausgewählten Beitrag zu bearbeiten und in einem kurzen Vortrag vorzustellen.</p> |
| <p>Literatur:</p> <p>abhängig vom jeweiligen Thema</p> |
| <p>Zugeordnete Lehrveranstaltungen:</p> <p>Seminar zu aktuellen Forschungsthemen der Mechatronik (Seminar) <i>*Veranstaltung wird online/digital abgehalten.*</i></p> |
| <p>Prüfung</p> <p>Vortrag und schriftliche Ausarbeitung Seminar</p> |

| | | |
|--|--|--|
| Modul INF-0318: Praktikum Simulation von autonomen Fahrzeugen (Vertiefung) <i>Practice Module on Simulation of Autonomous Vehicles</i> | | 8 ECTS/LP |
| Version 1.0.0 (seit WS19/20) Modulverantwortliche/r: Prof. Dr. Lars Mikelsons | | |
| Lernziele/Kompetenzen: Fachbezogene Kompetenzen: Nach der Teilnahme am Praktikum verstehen die Studierenden Problemstellungen höherer Komplexität auf dem Gebiet der Simulation von autonomen Fahrzeugen und verfügen dort über tiefere Fachkenntnisse und Fähigkeiten. Sie können Konzepte, Methoden, Verfahren, Techniken und Technologien des genannten Gebiets in Kleingruppen entwickeln und sind fähig, innovative Methoden bei der Lösung von Problemen anzuwenden. Dadurch ist es ihnen möglich, an die internationale Forschung anzuknüpfen und bestehende Konzepte umzusetzen. 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. | | |
| Schlüsselkompetenzen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken; Validierung von Simulationsergebnissen und Softwaremodulen; 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; | | |
| Arbeitsaufwand: Gesamt: 240 Std. 90 Std. Praktikum (Präsenzstudium) 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) | | |
| Voraussetzungen: Gute Programmierkenntnisse Hilfreich: Python, C++, ROS, Game Engines | | ECTS/LP-Bedingungen: Bestehen der Modulprüfung |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Praktikum Simulation von autonomen Fahrzeugen (Vertiefung) Lehrformen: Praktikum Sprache: Deutsch / Englisch SWS: 6 ECTS/LP: 8.0 | | |
| Inhalte: Im Praktikum bearbeiten die Studierenden in Kleingruppen anwendungsorientierte Aufgaben zur Simulation von autonomen Fahrzeugen. Dabei dürfen Sie einzelne Softwaremodule eines virtuellen Fahrzeugs selbst implementieren und validieren. | | |
| Zugeordnete Lehrveranstaltungen: | | |

Praktikum Simulation von autonomen Fahrzeugen (Vertiefung) (Praktikum)

Veranstaltung wird als Hybrid/gemischt abgehalten.

Anmeldung bis 30. Oktober 2020. Teilnehmer werden über Zulassung zum Praktikum dann informiert.

Prüfung

Praktikum Simulation von autonomen Fahrzeugen

Portfolioprüfung

| | | |
|--|---|--|
| Modul INF-0319: Praktikum Interdisziplinäres Projekt Ingenieurinformatik <i>Interdisciplinary Project Engineering Informatics</i> | | 6 ECTS/LP |
| Version 1.0.0 (seit WS19/20) Modulverantwortliche/r: Prof. Dr.-Ing. Johannes Schilp Prof. Dr.-Ing. Lars Mikelsons, Prof. Dr.-Ing. Christoph Ament | | |
| <p>Lernziele/Kompetenzen: Die Studierenden beschäftigen sich mit einer fächerübergreifenden Aufgabenstellung aus dem Bereich der Produktionsinformatik, Regelungstechnik und Mechatronik. Sie setzen komplexe Verfahren und Techniken ein, die teilweise in den einzelnen Vorlesungen bereits theoretisch behandelt wurden. Das praxisnahe Projekt orientiert sich an einer Studenten-Challenge, wie beispielweise der Sioux Mechatronics Trophy oder dem James Dyson Award, und wird in Kleingruppen bearbeitet. Der Anwendungsfall erfordert die Bewertung und Übertragung der Konzepte und Methoden sowie deren interdisziplinäre Kombination.</p> <p>Der zeitliche Ablauf dieses Praktikums wird in Digicampus bekannt gegeben, ebenso wie die spezifische Aufgabenstellung der Challenge einschließlich der Voraussetzungen und der Lehrstuhleteiligungen.</p> <p>Schlüsselqualifikationen: Team- und Kommunikationsfähigkeit, strukturiertes und gewissenhaftes Arbeiten, anwendungsorientierte Problemlösung, Ergebnisbewertung und -dokumentation, Abwägen von Lösungsansätzen, Fertigkeit zum logischen, analytischen und konzeptionellen Denken, Abstraktionsfähigkeit</p> | | |
| <p>Arbeitsaufwand: Gesamt: 180 Std. 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Praktikum (Präsenzstudium)</p> | | |
| Voraussetzungen: keine | | ECTS/LP-Bedingungen: Bestehen der Modulprüfung |
| Angebotshäufigkeit: jedes Sommersemester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Moduleile | | |
| <p>Modulteil: Praktikum Interdisziplinäres Projekt Ingenieurinformatik Lehrformen: Praktikum Sprache: Deutsch SWS: 4 ECTS/LP: 6.0</p> | | |
| Inhalte: Die Studenten bearbeiten in Kleingruppen anwendungsorientierte Aufgaben zu Themenbereichen im industriellen Umfeld. | | |
| Literatur: wird im jeweiligen Semester bekannt gegeben. | | |
| <p>Prüfung Praktikum Interdisziplinäres Projekt Ingenieurinformatik praktische Prüfung</p> | | |

| | | |
|--|---|--|
| Modul INF-0333: Konzepte autonomer mechatronischer Systeme <i>Concepts of autonomous vehicles</i> | | 5 ECTS/LP |
| Version 1.1.0 (seit SoSe20) Modulverantwortliche/r: Prof. Dr. Lars Mikelsons | | |
| <p>Lernziele/Kompetenzen:</p> <p>Lernziele/Kompetenzen:</p> <p>Fachbezogene Kompetenzen:</p> <p>Nach der erfolgreichen Teilnahme an diesem Modul können die Studierenden fortgeschrittene Methoden der Planung anwenden, um mechatronische Systeme in Lage zu versetzen autonom zu agieren. Sie kennen Methoden zur Umsetzung des Planungsergebnisses und können die anwenden.</p> <p>Methodische Kompetenzen:</p> <p>Die Studierenden sind in der Lage die unterschiedlichen Aspekte von Algorithmen und Methoden für die Planung zu bewerten. Sie können darüber hinaus die Folgen von dem Einsatz der Methoden abschätzen und somit Systemverhalten analysieren. Sie können basierend auf einer Beschreibung des physikalischen Systems ein objektorientiertes mathematisches Modell aufstellen.</p> <p>Fachübergreifende Kompetenzen:</p> <p>Die Studierenden sind in der Lage gewisse ingenieurwissenschaftliche Fragestellungen mit Hilfe von Informationstechnologie zu lösen. Sie können Abstraktionen physikalischer Systeme erzeugen und diese in informationstechnologische Zusammenhänge einbinden.</p> <p>Schlüsselkompetenzen:</p> <p>Fortgeschrittene Fertigkeit ingenieurwissenschaftlicher Fragestellungen, insbesondere aus dem Bereich des Entwurfs autonomer mechatronischer Systeme zu beantworten; Beurteilung und Analyse von mathematischen Methoden</p> | | |
| <p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std.</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>30 Std. Übung (Präsenzstudium)</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: Konzepte autonomer mechatronischer Systeme (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Dozenten: Prof. Dr. Lars Mikelsons</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p> | | |

Inhalte:

Inhalt:

In der Vorlesung werden moderne Konzepte von autonomen mechatronischen Systemen behandelt. Ausgehend vom see-think-act Zyklus lernen die Studierenden die Herausforderungen kennen, damit ein mechatronisches System autonom agieren kann. Der Fokus der Veranstaltung liegt dabei auf der Planung (think) und der Ausführung (act). Die Studierenden lernen sowohl die theoretischen Hintergründe entsprechender Methoden als auch deren praktische Umsetzung in der Übung.

Literatur:

“Planning” von LaValle

wird in der Vorlesung bekannt gegeben

Modulteil: Konzepte autonomer mechatronischer Systeme (Übung)

Lehrformen: Übung

Sprache: Deutsch

Angebotshäufigkeit: jedes Wintersemester

SWS: 2

Inhalte:

Wiederholung und Vertiefung der Lehrinhalte aus der Vorlesung mithilfe von Übungen und Praxisbeispielen

Prüfung

Konzepte autonomer mechatronischer Systeme

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

Die Wiederholungsprüfung findet zu Beginn des folgenden Semesters statt.

| | | |
|---|---|---|
| Modul INF-0003: Masterarbeit <i>Master's Thesis</i> | | 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 tiefere 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: Gesamt: 900 Std. 900 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: 4. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 0 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Moduleile | | |
| Modulteil: Masterarbeit | | |
| Sprache: Deutsch | | |
| Inhalte: entsprechend dem gewählten Thema | | |

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 <i>Introduction to Scientific Work</i> | | 0 ECTS/LP |
| Version 1.0.0 (seit WS15/16) 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 <i>*Veranstaltung wird online/digital abgehalten.*</i> | | |
| Anleitung zu wissenschaftlichen Arbeiten <i>*Veranstaltung wird online/digital abgehalten.*</i> | | |
| Anleitung zu wissenschaftlichen Arbeiten Software- und Systems Engineering <i>*Veranstaltung wird online/digital abgehalten.*</i> | | |

| | | |
|---|-----------------------------------|---|
| Modul INF-0222: Oberseminar Informatik <i>Graduate Seminar Computer Science</i> | | 0 ECTS/LP |
| Version 1.0.0 (seit WS15/16) 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 |
| Modulteil: Oberseminar Informatik Lehrformen: Seminar Sprache: Deutsch SWS: 2 |
| Zugeordnete Lehrveranstaltungen: Oberseminar Datenbanken und Informationssysteme <i>*Veranstaltung wird online/digital abgehalten.*</i> Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren direkt einen Mitarbeiter, dessen Forschung Sie interessiert: https://www.uni-augsburg.de/de/fakultaet/fai/informatik/prof/dbis/ Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte! Oberseminar Embedded Intelligence for Health Care and Wellbeing <i>*Veranstaltung wird online/digital abgehalten.*</i> Oberseminar Embedded Systems <i>*Veranstaltung wird online/digital abgehalten.*</i> Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren direkt einen Mitarbeiter, dessen Forschung Sie interessiert: www.es-augsburg.de Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte. Oberseminar Human-Centered Multimedia <i>*Veranstaltung wird online/digital abgehalten.*</i> Oberseminar IT-Infrastrukturen für die Translationale Medizinische Forschung <i>*Veranstaltung wird online/digital abgehalten.*</i> |

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen und kontaktieren direkt einen Mitarbeiter, dessen Forschung Sie interessiert: <https://www.uni-augsburg.de/de/fakultaet/fai/informatik/prof/misit/> Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Oberseminar Multimedia Computing

Veranstaltung wird online/digital abgehalten.

Oberseminar Organic Computing

Veranstaltung wird online/digital abgehalten.

Weiterführende Themen aus dem Bereich Organic Computing

Oberseminar Resource Aware Algorithmics

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge und Arbeitsbesprechungen zu Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unter Lehre (siehe <https://www.uni-augsburg.de/de/fakultaet/fai/informatik/prof/raa/studium-und-lehre/>) unsere Informationen zu Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen (unter Lehrveranstaltungen) an und kontaktieren direkt einen Mitarbeiter. Die alleinige Teilnahme am Oberseminar ist nicht möglich. Forschungsthemen des Lehrstuhls: Approximationsalgorithmen, Online Algorithmen, Algorithmen für Big Data

Oberseminar Software- und Systems Engineering

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen am Lehrstuhl für Softwaretechnik zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren Sie direkt einen Mitarbeiter, dessen Forschung Sie interessiert: <https://www.uni-augsburg.de/de/fakultaet/fai/isse/prof/swtse/teaching/students/> Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Oberseminar Theoretische Informatik

Veranstaltung wird online/digital abgehalten.

Oberseminar zu Grundlagen Reaktiver Systeme

Veranstaltung wird online/digital abgehalten.

Oberseminar zu Lehrprofessur für Informatik

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge und Arbeitsbesprechungen zu Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unter Lehre (siehe <https://www.uni-augsburg.de/de/fakultaet/fai/informatik/prof/educ-inf/lehre/>) unsere Informationen zu Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen (unter Lehrveranstaltungen) an und kontaktieren direkt einen Mitarbeiter. Die alleinige Teilnahme am Oberseminar ist nicht möglich. Forschungsthemen des Lehrstuhls: Petri Nets, Process Mining, Concurrent Systems

Oberseminar zu Softwaremethodik für verteilte Systeme

Veranstaltung wird online/digital abgehalten.

Oberseminar zur Mechatronik

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn du ein solches Modul am Lehrstuhl für Mechatronik belegen willst, komm doch einfach auf uns zu (persönlich, Mail, zoom, ...). Wir freuen uns mit dir unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten zu diskutieren. Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Oberseminar zur Produktionsinformatik

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren direkt einen Mitarbeiter, dessen Forschung Sie interessiert: <https://www.uni-augsburg.de/de/fakultaet/fai/informatik/prof/pi/lehre/fm-pm-seminar/> Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Oberseminar zur Regelungstechnik

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren direkt einen Mitarbeiter, dessen Forschung Sie interessiert: <https://www.informatik.uni-augsburg.de/lehrstuehle/rt/> Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte!

Oberseminar zur biomedizinischen Informatik

Veranstaltung wird online/digital abgehalten.

Das Oberseminar fasst Vorträge von Abschlussarbeiten und Praxis-/Forschungs-/Projektmodulen zusammen. Wenn Sie ein solches Modul belegen wollen, schauen Sie sich auf unserer Lehrstuhlhomepage unsere Forschungsthemen bzw. Themenvorschläge für studentische Arbeiten an und kontaktieren direkt einen Mitarbeiter, dessen Forschung Sie interessiert: www.uni-augsburg.de/de/fakultaet/fai/informatik/prof/bioinf/ Die alleinige Teilnahme am Oberseminar ist freiwillig und gibt keine ECTS-Punkte.