

Modulhandbuch

M.Sc. Ingenieurinformatik, PO 2016

Fakultät für Angewandte Informatik

Sommersemester 2017

Liebe Studierenden,

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

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

Im M.Sc. Ingenieurinformatik gibt es außerdem noch folgende Neuerungen:

1. Das Modul INF-0239: Projektmodul Industrie 4.0 wurde umbenannt zu Projektmodul Produktionsinformatik und trägt daher die neue Modulsignatur INF-0244
2. Der Bereich Mechatronik und Produktion wurde deutlich ausgebaut mit folgenden Modulen:
 - INF-0245: Intelligent vernetzte Produktion
 - INF-0246: Seminar Industrie 4.0
 - INF-0247: Praktikum für Produktionsinformatik (Vertiefung)
 - INF-0249: Seminar zu speziellen Kapiteln der Regelungstechnik

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

fsinfo@informatik.uni-augsburg.de

und persönlich im Raum 1007N.

Viele Grüße,

Euer Modulhandbuch-Beauftragter

Martin Frieb

| ID | Modul | Semester | ECTS | SWS | Prüfung |
|----------------------------------|---|----------------------|-----------|------------------------|--------------------------------|
| M.Sc. Ingenieurinformatik | | | | | |
| 1 | Modulgruppe: Software und Systems Engineering | | 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-0032 | Automotive Software Engineering | unregelmäßig | 5 | 3 Vorlesung | Mündliche Prüfung 30Minuten |
| INF-0037 | Praktikum Automotive Software Engineering | jedes Semester | 6 | 6 Praktikum | Mündliche Prüfung 30Minuten |
| INF-0038 | Praktikum Avionic Software Engineering | unregelmäßig | 6 | 6 Praktikum | Mündliche Prüfung 30Minuten |
| INF-0040 | Seminar Grundlagen des Software Engineering für Automotive Systems (MA) | jedes Semester | 4 | 2 Seminar | Seminar |
| INF-0041 | Seminar Grundlagen des Software Engineering für Avionic Systems (MA) | jedes Semester | 4 | 2 Seminar | Seminar |
| INF-0042 | Projektmodul Softwaremethodiken für verteilte Systeme | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0129 | Softwaretechnik II | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Klausur 90Minuten |
| INF-0131 | Software- und Systemsicherheit | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Mündliche Prüfung 45Minuten |

| | | | | | |
|----------|--|---------------------------------|----|------------------------|---------------------------------|
| INF-0136 | Seminar Software- und Systems Engineering (Master) | jedes Semester | 4 | 2 Seminar | Seminar |
| INF-0137 | Projektmodul Software- und Systems Engineering | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0189 | Qualitätssicherung im Software Engineering | unregelmäßig (i. d. R. im SoSe) | 5 | 2 Vorlesung 2 Übung | Klausur 90Minuten |
| INF-0232 | Seminar Medical Information Sciences (MA) | jedes Semester | 4 | 2 Seminar | Seminar |
| INF-0233 | Industrierobotik | jedes Semester | 8 | 2 Vorlesung 4 Übung | praktische Prüfung 45Minuten |
| INF-0234 | Flugrobotik | unregelmäßig (i. d. R. im SoSe) | 5 | 2 Vorlesung 2 Übung | praktische Prüfung 45Minuten |
| INF-0235 | Software für Industrie 4.0 | jedes Wintersemester | 5 | 2 Vorlesung 2 Übung | Mündliche Prüfung 45Minuten |
| INF-0248 | Kollaborative Robotik | unregelmäßig | 8 | 2 Vorlesung 4 Übung | praktische Prüfung 60Minuten |

2 Modulgruppe: Technische Informatik und Adaptive Systeme 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 | jedes Sommersemester | 5 | 2 Vorlesung 2 Übung | Mündliche Prüfung 30Minuten |
| INF-0067 | Peer-to-Peer und Cloud Computing | jedes Wintersemester | 5 | 2 Vorlesung 2 Übung | Mündliche Prüfung 30Minuten |
| INF-0069 | Weiterführende Betriebssystemkonzepte | unregelmäßig | 8 | 2 Vorlesung 4 Übung | Klausur 90Minuten |
| INF-0070 | Seminar Organic Computing | jedes Wintersemester | 4 | 2 Seminar | Seminar |
| INF-0071 | Seminar Naturanaloge Algorithmen und Multiagentensysteme | jedes Sommersemester | 4 | 2 Seminar | Seminar |
| INF-0072 | Projektmodul Organic Computing | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0084 | Seminar Next Generation Networks | jedes Sommersemester | 4 | 2 Seminar | Seminar |
| INF-0085 | Projektmodul Kommunikationssysteme | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0088 | Bayesian Networks | jedes Sommersemester | 5 | 2 Vorlesung 2 Übung | Klausur 90Minuten |
| INF-0092 | Multimedia II: Machine Learning and Computer Vision | jedes Sommersemester | 8 | 4 Vorlesung 2 Übung | Klausur 120Minuten |
| INF-0093 | Probabilistic Robotics | jedes Wintersemester | 5 | 2 Vorlesung 2 Übung | Klausur 90Minuten |

| | | | | | |
|----------|--|------------------------------------|---|------------------------|--------------------------------|
| INF-0094 | Maschinelles Lernen | unregelmäßig | 5 | 2 Vorlesung 2 Übung | Klausur 90Minuten |
| INF-0133 | Selbstorganisierende, adaptive Systeme | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Mündliche Prüfung 45Minuten |
| INF-0145 | Mikrorechner-technik und Echtzeitsysteme | jedes Wintersemester | 6 | 3 Vorlesung 1 Übung | Klausur 60Minuten |
| INF-0146 | Cyber-Physical Systems | wird nicht mehr angeboten! | 6 | 3 Vorlesung 1 Übung | Klausur 90Minuten |
| INF-0147 | Prozessorarchitektur | jedes Sommersemester | 5 | 2 Vorlesung 2 Übung | Klausur 60Minuten |
| INF-0148 | Entwurf und Analyse fehlertolerierender Rechensysteme | jedes Wintersemester | 5 | 2 Vorlesung 2 Übung | Klausur 60Minuten |
| INF-0149 | Praktikum Eingebettete Systeme | jedes Sommersemester | 5 | 4 Praktikum | Praktikum |
| INF-0150 | Hardware-Entwurf | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Praktikum |
| INF-0151 | Praktikum Multicore-Programmierung | wurde ersetzt durch INF-0216 | 5 | 4 Praktikum | Praktikum |
| INF-0152 | Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen | jedes Sommersemester | 4 | 2 Seminar | Seminar |
| INF-0153 | Seminar Safety-Critical Systems | jedes Wintersemester | 4 | 2 Seminar | Seminar |

| | | | | | |
|----------|--|---------------------------------------|-----------|------------------------|--|
| INF-0154 | Projektmodul Systemnahe Informatik und Kommunikationssysteme | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0176 | Digital Signal Processing II | wird nicht mehr angeboten! | 6 | 4 Vorlesung | Klausur 100Minuten |
| INF-0180 | Computational Intelligence | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Mündliche Prüfung |
| INF-0182 | Praktikum Multimodale Echtzeitsignalverarbeitung | jedes Sommersemester | 8 | 6 Praktikum | Projektarbeit |
| INF-0216 | Vertiefte Multicore-Programmierung | jedes Sommersemester | 8 | 2 Vorlesung 4 Übung | Mündliche Prüfung (Dauer: 30-45 Minuten) Stunden |
| INF-0251 | Seminar Artificial Intelligence | in der Regel mind. 1x pro Studienjahr | 4 | 2 Seminar | Seminar Stunden |
| 3 | <i>Modulgruppe: Wahlbereich Software und Systems Engineering / Technische Informatik und Adaptive Systeme</i> | | 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-0032 | Automotive Software Engineering | unregelmäßig | 5 | 3 Vorlesung | Mündliche Prüfung 30Minuten |
| INF-0037 | Praktikum Automotive Software Engineering | jedes Semester | 6 | 6 Praktikum | Mündliche Prüfung 30Minuten |

| | | | | | |
|----------|---|----------------------|----|------------------------|--------------------------------|
| INF-0038 | Praktikum Avionic Software Engineering | unregelmäßig | 6 | 6 Praktikum | Mündliche Prüfung 30Minuten |
| INF-0040 | Seminar Grundlagen des Software Engineering für Automotive Systems (MA) | jedes Semester | 4 | 2 Seminar | Seminar |
| INF-0041 | Seminar Grundlagen des Software Engineering für Avionic Systems (MA) | jedes Semester | 4 | 2 Seminar | Seminar |
| INF-0042 | Projektmodul Softwaremethodiken für verteilte Systeme | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0066 | Organic Computing II | jedes Sommersemester | 5 | 2 Vorlesung 2 Übung | Mündliche Prüfung 30Minuten |
| INF-0067 | Peer-to-Peer und Cloud Computing | jedes Wintersemester | 5 | 2 Vorlesung 2 Übung | Mündliche Prüfung 30Minuten |
| INF-0069 | Weiterführende Betriebssystemkonzepte | unregelmäßig | 8 | 2 Vorlesung 4 Übung | Klausur 90Minuten |
| INF-0070 | Seminar Organic Computing | jedes Wintersemester | 4 | 2 Seminar | Seminar |
| INF-0071 | Seminar Naturanaloge Algorithmen und Multiagentensysteme | jedes Sommersemester | 4 | 2 Seminar | Seminar |
| INF-0072 | Projektmodul Organic Computing | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0084 | Seminar Next Generation Networks | jedes Sommersemester | 4 | 2 Seminar | Seminar |

| | | | | | |
|----------|---|-------------------------|----|------------------------|--------------------------------|
| INF-0085 | Projektmodul Kommunikationssysteme | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0088 | Bayesian Networks | jedes Sommersemester | 5 | 2 Vorlesung 2 Übung | Klausur 90Minuten |
| INF-0092 | Multimedia II: Machine Learning and Computer Vision | jedes Sommersemester | 8 | 4 Vorlesung 2 Übung | Klausur 120Minuten |
| INF-0093 | Probabilistic Robotics | jedes Wintersemester | 5 | 2 Vorlesung 2 Übung | Klausur 90Minuten |
| INF-0094 | Maschinelles Lernen | unregelmäßig | 5 | 2 Vorlesung 2 Übung | Klausur 90Minuten |
| INF-0129 | Softwaretechnik II | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Klausur 90Minuten |
| INF-0131 | Software- und Systemsicherheit | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Mündliche Prüfung 45Minuten |
| INF-0133 | Selbstorganisierende, adaptive Systeme | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Mündliche Prüfung 45Minuten |
| INF-0136 | Seminar Software- und Systems Engineering (Master) | jedes Semester | 4 | 2 Seminar | Seminar |
| INF-0137 | Projektmodul Software- und Systems Engineering | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0145 | Mikrorechner- und Echtzeitsysteme | jedes Wintersemester | 6 | 3 Vorlesung 1 Übung | Klausur 60Minuten |

| | | | | | |
|----------|---|------------------------------------|----|------------------------|-----------------------|
| INF-0146 | Cyber-Physical Systems | wird nicht mehr angeboten! | 6 | 3 Vorlesung 1 Übung | Klausur 90Minuten |
| INF-0147 | Prozessorarchitektur | jedes Sommersemester | 5 | 2 Vorlesung 2 Übung | Klausur 60Minuten |
| INF-0148 | Entwurf und Analyse fehlertolerierender Rechensysteme | jedes Wintersemester | 5 | 2 Vorlesung 2 Übung | Klausur 60Minuten |
| INF-0149 | Praktikum Eingebettete Systeme | jedes Sommersemester | 5 | 4 Praktikum | Praktikum |
| INF-0150 | Hardware-Entwurf | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Praktikum |
| INF-0151 | Praktikum Multicore-Programmierung | wurde ersetzt durch INF-0216 | 5 | 4 Praktikum | Praktikum |
| INF-0152 | Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen | jedes Sommersemester | 4 | 2 Seminar | Seminar |
| INF-0153 | Seminar Safety-Critical Systems | jedes Wintersemester | 4 | 2 Seminar | Seminar |
| INF-0154 | Projektmodul Systemnahe Informatik und Kommunikationssysteme | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0176 | Digital Signal Processing II | wird nicht mehr angeboten! | 6 | 4 Vorlesung | Klausur 100Minuten |

| | | | | | |
|----------|--|---|-----------|------------------------|--|
| INF-0180 | Computational Intelligence | jedes Wintersemester | 8 | 2 Vorlesung 4 Übung | Mündliche Prüfung |
| INF-0182 | Praktikum Multimodale Echtzeitsignalverarbeitung | jedes Sommersemester | 8 | 6 Praktikum | Projektarbeit |
| INF-0189 | Qualitätssicherung im Software Engineering | unregelmäßig (i. d. R. im SoSe) | 5 | 2 Vorlesung 2 Übung | Klausur 90Minuten |
| INF-0216 | Vertiefte Multicore-Programmierung | jedes Sommersemester | 8 | 2 Vorlesung 4 Übung | Mündliche Prüfung (Dauer: 30-45 Minuten) Stunden |
| INF-0232 | Seminar Medical Information Sciences (MA) | jedes Semester | 4 | 2 Seminar | Seminar |
| INF-0233 | Industrierobotik | jedes Semester | 8 | 2 Vorlesung 4 Übung | praktische Prüfung 45Minuten |
| INF-0234 | Flugrobotik | unregelmäßig (i. d. R. im SoSe) | 5 | 2 Vorlesung 2 Übung | praktische Prüfung 45Minuten |
| INF-0235 | Software für Industrie 4.0 | jedes Wintersemester | 5 | 2 Vorlesung 2 Übung | Mündliche Prüfung 45Minuten |
| INF-0248 | Kollaborative Robotik | unregelmäßig | 8 | 2 Vorlesung 4 Übung | praktische Prüfung 60Minuten |
| INF-0251 | Seminar Artificial Intelligence | in der Regel mind. 1x pro Studienjahr | 4 | 2 Seminar | Seminar Stunden |
| 4 | Modulgruppe: Mechatronik und Produktion | | 36 | | |

| "36 Leistungspunkte aus der Modulgruppe Mechatronik und Produktion" | | | | | |
|---|--|-------------------------|-----------|------------------------|---|
| INF-0236 | Digitale Regelsysteme | jedes Wintersemester | 6 | 3 Vorlesung 2 Übung | Schriftlich-Mündliche Prüfung 30Minuten |
| INF-0237 | Projektmodul Regelungstechnik | nach Bedarf | 10 | - | Praktikum keine Einheit gewählt |
| INF-0238 | Digitale Fabrik | jedes Wintersemester | 6 | 3 Vorlesung 2 Übung | Schriftlich-Mündliche Prüfung 30Minuten |
| INF-0244 | Projektmodul Produktionsinformatik | nach Bedarf | 10 | 1 Praktikum | Praktikum |
| INF-0245 | Intelligent vernetzte Produktion | jedes Sommersemester | 5 | 2 Vorlesung 2 Übung | Schriftlich-Mündliche Prüfung 90Minuten |
| INF-0246 | Seminar Industrie 4.0 | jedes Sommersemester | 4 | 2 Seminar | Seminar keine Einheit gewählt |
| INF-0247 | Praktikum für Produktionsinformatik (Vertiefung) | jedes Sommersemester | 6 | 4 Praktikum | Praktikum |
| INF-0249 | Seminar zu speziellen Kapiteln der Regelungstechnik | jedes Sommersemester | 4 | 2 | Seminar (Vortrag und schriftliche Ausarbeitung) keine Einheit gewählt |
| 5 | Modulgruppe: Abschlussarbeit | | 30 | | |
| "30 Leistungspunkte im Rahmen der Masterarbeit" | | | | | |
| INF-0003 | Masterarbeit | nach Bedarf | 30 | 1 | Masterarbeit |

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

Übersicht nach Modulgruppen

1) 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-0032: Automotive Software Engineering (5 ECTS/LP, Wahlpflicht)..... | 7 |
| INF-0037: Praktikum Automotive Software Engineering (6 ECTS/LP, Wahlpflicht)..... | 8 |
| INF-0038: Praktikum Avionic Software Engineering (6 ECTS/LP, Wahlpflicht)..... | 10 |
| INF-0040: Seminar Grundlagen des Software Engineering für Automotive Systems (MA) (4 ECTS/LP, Wahlpflicht)..... | 12 |
| INF-0041: Seminar Grundlagen des Software Engineering für Avionic Systems (MA) (4 ECTS/LP, Wahlpflicht)..... | 13 |
| INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme (10 ECTS/LP, Wahlpflicht)..... | 14 |
| INF-0129: Softwaretechnik II (8 ECTS/LP, Wahlpflicht)..... | 15 |
| INF-0131: Software- und Systemsicherheit (8 ECTS/LP, Wahlpflicht)..... | 17 |
| INF-0136: Seminar Software- und Systems Engineering (Master) (4 ECTS/LP, Wahlpflicht)..... | 19 |
| INF-0137: Projektmodul Software- und Systems Engineering (10 ECTS/LP, Wahlpflicht)..... | 21 |
| INF-0189: Qualitätssicherung im Software Engineering (5 ECTS/LP, Wahlpflicht)..... | 22 |
| INF-0232: Seminar Medical Information Sciences (MA) (4 ECTS/LP, Wahlpflicht)..... | 24 |
| 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 |

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)..... | 33 |
| INF-0067: Peer-to-Peer und Cloud Computing (5 ECTS/LP, Wahlpflicht)..... | 35 |
| INF-0069: Weiterführende Betriebssystemkonzepte (8 ECTS/LP, Wahlpflicht)..... | 37 |

| | |
|---|----|
| INF-0070: Seminar Organic Computing (4 ECTS/LP, Wahlpflicht)..... | 39 |
| INF-0071: Seminar Naturanaloge Algorithmen und Multiagentensysteme (4 ECTS/LP, Wahlpflicht)..... | 40 |
| INF-0072: Projektmodul Organic Computing (10 ECTS/LP, Wahlpflicht)..... | 41 |
| INF-0084: Seminar Next Generation Networks (4 ECTS/LP, Wahlpflicht)..... | 42 |
| INF-0085: Projektmodul Kommunikationssysteme (10 ECTS/LP, Wahlpflicht)..... | 43 |
| INF-0088: Bayesian Networks (5 ECTS/LP, Wahlpflicht)..... | 44 |
| INF-0092: Multimedia II: Machine Learning and Computer Vision (8 ECTS/LP, Wahlpflicht)..... | 46 |
| INF-0093: Probabilistic Robotics (5 ECTS/LP, Wahlpflicht)..... | 48 |
| INF-0094: Maschinelles Lernen (5 ECTS/LP, Wahlpflicht)..... | 50 |
| INF-0133: Selbstorganisierende, adaptive Systeme (8 ECTS/LP, Wahlpflicht)..... | 52 |
| INF-0145: Mikrorechnertechnik und Echtzeitsysteme (6 ECTS/LP, Wahlpflicht)..... | 54 |
| INF-0146: Cyber-Physical Systems (6 ECTS/LP, Wahlpflicht)..... | 56 |
| INF-0147: Prozessorarchitektur (5 ECTS/LP, Wahlpflicht)..... | 58 |
| INF-0148: Entwurf und Analyse fehlertolerierender Rechensysteme (5 ECTS/LP, Wahlpflicht)..... | 60 |
| INF-0149: Praktikum Eingebettete Systeme (5 ECTS/LP, Wahlpflicht)..... | 62 |
| INF-0150: Hardware-Entwurf (8 ECTS/LP, Wahlpflicht)..... | 63 |
| INF-0151: Praktikum Multicore-Programmierung (5 ECTS/LP, Wahlpflicht)..... | 65 |
| INF-0152: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen (4 ECTS/LP, Wahlpflicht)..... | 66 |
| INF-0153: Seminar Safety-Critical Systems (4 ECTS/LP, Wahlpflicht)..... | 67 |
| INF-0154: Projektmodul Systemnahe Informatik und Kommunikationssysteme (10 ECTS/LP, Wahlpflicht)..... | 68 |
| INF-0176: Digital Signal Processing II (6 ECTS/LP, Wahlpflicht)..... | 69 |
| INF-0180: Computational Intelligence (8 ECTS/LP, Wahlpflicht)..... | 70 |
| INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung (8 ECTS/LP, Wahlpflicht)..... | 72 |
| INF-0216: Vertiefte Multicore-Programmierung (8 ECTS/LP, Wahlpflicht)..... | 74 |
| INF-0251: Seminar Artificial Intelligence (4 ECTS/LP, Wahlpflicht)..... | 76 |

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-0032: Automotive Software Engineering (5 ECTS/LP, Wahlpflicht)..... | 78 |
| INF-0037: Praktikum Automotive Software Engineering (6 ECTS/LP, Wahlpflicht)..... | 79 |
| INF-0038: Praktikum Avionic Software Engineering (6 ECTS/LP, Wahlpflicht)..... | 81 |
| INF-0040: Seminar Grundlagen des Software Engineering für Automotive Systems (MA) (4 ECTS/LP, Wahlpflicht)..... | 83 |
| INF-0041: Seminar Grundlagen des Software Engineering für Avionic Systems (MA) (4 ECTS/LP, Wahlpflicht)..... | 84 |
| INF-0042: Projektmodul Softwaremethodiken für verteilte Systeme (10 ECTS/LP, Wahlpflicht)..... | 85 |
| INF-0066: Organic Computing II (5 ECTS/LP, Wahlpflicht)..... | 86 |
| INF-0067: Peer-to-Peer und Cloud Computing (5 ECTS/LP, Wahlpflicht)..... | 88 |
| INF-0069: Weiterführende Betriebssystemkonzepte (8 ECTS/LP, Wahlpflicht)..... | 90 |
| INF-0070: Seminar Organic Computing (4 ECTS/LP, Wahlpflicht)..... | 92 |
| INF-0071: Seminar Natural analoge Algorithmen und Multiagentensysteme (4 ECTS/LP, Wahlpflicht)..... | 93 |
| INF-0072: Projektmodul Organic Computing (10 ECTS/LP, Wahlpflicht)..... | 94 |
| INF-0084: Seminar Next Generation Networks (4 ECTS/LP, Wahlpflicht)..... | 95 |
| INF-0085: Projektmodul Kommunikationssysteme (10 ECTS/LP, Wahlpflicht)..... | 96 |
| INF-0088: Bayesian Networks (5 ECTS/LP, Wahlpflicht)..... | 97 |
| INF-0092: Multimedia II: Machine Learning and Computer Vision (8 ECTS/LP, Wahlpflicht)..... | 99 |
| INF-0093: Probabilistic Robotics (5 ECTS/LP, Wahlpflicht)..... | 101 |
| INF-0094: Maschinelles Lernen (5 ECTS/LP, Wahlpflicht)..... | 103 |
| INF-0129: Softwaretechnik II (8 ECTS/LP, Wahlpflicht)..... | 105 |
| INF-0131: Software- und Systemsicherheit (8 ECTS/LP, Wahlpflicht)..... | 107 |
| INF-0133: Selbstorganisierende, adaptive Systeme (8 ECTS/LP, Wahlpflicht)..... | 109 |
| INF-0136: Seminar Software- und Systems Engineering (Master) (4 ECTS/LP, Wahlpflicht)..... | 111 |
| INF-0137: Projektmodul Software- und Systems Engineering (10 ECTS/LP, Wahlpflicht)..... | 113 |
| INF-0145: Mikrorechnerarchitektur und Echtzeitsysteme (6 ECTS/LP, Wahlpflicht)..... | 114 |
| INF-0146: Cyber-Physical Systems (6 ECTS/LP, Wahlpflicht)..... | 116 |
| INF-0147: Prozessorarchitektur (5 ECTS/LP, Wahlpflicht)..... | 118 |
| INF-0148: Entwurf und Analyse fehlertolerierender Rechensysteme (5 ECTS/LP, Wahlpflicht)..... | 120 |
| INF-0149: Praktikum Eingebettete Systeme (5 ECTS/LP, Wahlpflicht)..... | 122 |
| INF-0150: Hardware-Entwurf (8 ECTS/LP, Wahlpflicht)..... | 123 |

| | |
|---|-----|
| INF-0151: Praktikum Multicore-Programmierung (5 ECTS/LP, Wahlpflicht)..... | 125 |
| INF-0152: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen (4 ECTS/LP, Wahlpflicht)..... | 126 |
| INF-0153: Seminar Safety-Critical Systems (4 ECTS/LP, Wahlpflicht)..... | 127 |
| INF-0154: Projektmodul Systemnahe Informatik und Kommunikationssysteme (10 ECTS/LP, Wahlpflicht)..... | 128 |
| INF-0176: Digital Signal Processing II (6 ECTS/LP, Wahlpflicht)..... | 129 |
| INF-0180: Computational Intelligence (8 ECTS/LP, Wahlpflicht)..... | 130 |
| INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung (8 ECTS/LP, Wahlpflicht)..... | 132 |
| INF-0189: Qualitätssicherung im Software Engineering (5 ECTS/LP, Wahlpflicht)..... | 134 |
| INF-0216: Vertiefte Multicore-Programmierung (8 ECTS/LP, Wahlpflicht)..... | 136 |
| INF-0232: Seminar Medical Information Sciences (MA) (4 ECTS/LP, Wahlpflicht)..... | 138 |
| INF-0233: Industrierobotik (8 ECTS/LP, Wahlpflicht)..... | 139 |
| INF-0234: Flugrobotik (5 ECTS/LP, Wahlpflicht)..... | 141 |
| INF-0235: Software für Industrie 4.0 (5 ECTS/LP, Wahlpflicht)..... | 143 |
| INF-0248: Kollaborative Robotik (8 ECTS/LP, Wahlpflicht)..... | 145 |
| INF-0251: Seminar Artificial Intelligence (4 ECTS/LP, Wahlpflicht)..... | 147 |

d) Mechatronik und Produktion ECTS: 36

"36 Leistungspunkte aus der Modulgruppe Mechatronik und Produktion"

| | |
|---|-----|
| INF-0236: Digitale Regelsysteme (6 ECTS/LP, Wahlpflicht)..... | 149 |
| INF-0237: Projektmodul Regelungstechnik (10 ECTS/LP, Wahlpflicht)..... | 152 |
| INF-0238: Digitale Fabrik (6 ECTS/LP, Wahlpflicht)..... | 153 |
| INF-0244: Projektmodul Produktionsinformatik (10 ECTS/LP, Wahlpflicht)..... | 155 |
| INF-0245: Intelligent vernetzte Produktion (5 ECTS/LP, Wahlpflicht)..... | 156 |
| INF-0246: Seminar Industrie 4.0 (4 ECTS/LP, Wahlpflicht)..... | 158 |
| INF-0247: Praktikum für Produktionsinformatik (Vertiefung) (6 ECTS/LP, Wahlpflicht)..... | 159 |
| INF-0249: Seminar zu speziellen Kapiteln der Regelungstechnik (4 ECTS/LP, Wahlpflicht)..... | 160 |

e) Abschlussarbeit ECTS: 30

"30 Leistungspunkte im Rahmen der Masterarbeit"

| | |
|---|-----|
| INF-0003: Masterarbeit (30 ECTS/LP, Pflicht)..... | 161 |
|---|-----|

2) Freiwillige Veranstaltungen

Inhaltsverzeichnis

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)..... | 162 |
| INF-0222: Oberseminar Informatik (0 ECTS/LP, Wahlfach)..... | 163 |

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

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

| |
|--|
| Prüfung Automotive Software Engineering (mündl. Prüfung) Mündliche Prüfung / Prüfungsdauer: 30 Minuten |
|--|

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

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

Prüfung

Praktikum Automotive Software Engineering

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

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

Inhalte:

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

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

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

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

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

Die erforderlichen Tätigkeiten sind:

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

Vorkenntnisse:

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

Literatur:

abhängig vom Thema

Zugeordnete Lehrveranstaltungen:

Praktikum Avionic Software Engineering (Praktikum)

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

Prüfung

Praktikum Avionic Software Engineering

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

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

| | | |
|--|---|---|
| Modul INF-0041: Seminar Grundlagen des Software Engineering für Avionic Systems (MA) | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer | | |
| <p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet des Avionic Software Engineerings selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p>Schlüsselqualifikationen: Erlernen von Präsentationstechniken; Abwägen von Lösungsansätzen</p> | | |
| <p>Arbeitsaufwand: 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 | |

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

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

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

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

Inhalte:

Agile Softwareentwicklung:

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

Refactoring

- Code Smells
- Prinzipien des objektorientierten Designs
- Wichtige Refactorings

Testen

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

Requirements Engineering

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

Literatur:

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

Modulteil: Softwaretechnik II (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Softwaretechnik II Klausur

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|--|---|---|
| Modul INF-0131: Software- und Systemsicherheit | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| Lernziele/Kompetenzen: Die Studierenden können Bedrohungsanalyse durchführen, kryptographische Protokolle entwickeln, Chipkarten programmieren und sicherheitskritische Systeme entwerfen. Sie können systematisch Bedrohungen für Softwaresysteme analysieren und deren Risiken bewerten. Sie können einen modellgetriebenen Entwicklungsprozess für sicherheitskritische Systeme anwenden. Sie sind in der Lage, fachliche Lösungskonzepte in Programme umzusetzen. Sie haben Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen. | | |
| Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis | | |
| 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: 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: <ul style="list-style-type: none"> • 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) | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage selbstständig ein wissenschaftlich anspruchsvolles Thema aus dem Gebiet der Softwaretechnik zu erarbeiten, geeignet in Schrift und Vortrag zu präsentieren und sachlich über Vorträge zu diskutieren. Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis | | |
| Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Software- und Systems Engineering (Master) Lehrformen: Seminar Sprache: Deutsch 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 Software- und Systems Engineering (Master) (Seminar) In dem Seminar werden aktuelle Themen aus dem Umfeld des Software- und Systems Engineering behandelt. Es stehen folgende Themen noch zur Auswahl: * Data-Race Detection am Beispiel RV-Predict (https://runtimeverification.com/predict/) oder JavaPathfinder (http://babelfish.arc.nasa.gov/trac/jpf) * Dynamic Frames zur Verifikation Heap-basierter Programme (siehe "Dynamic Frames: Support for Framing, Dependencies and Sharing Without Restrictions" von Ioannis T. Kassios) * Static Driver Verifier von Microsoft (siehe z.B. https://www.microsoft.com/en-us/research/project/slam/ und "SLAM and Static Driver Verifier: Technology Transfer of Formal Methods inside Microsoft" von Thomas Ball, Byron Cook, Vladimir Levin und Sriram K. Rajamani) * eigener Themenvorschlag aus dem Bereich der Nebenläufigkeit, formalen Methoden oder statischen Analyse sind ebenfalls herzlich willkommen bereits vergeben sind: * Software Transactional Memory am Beispiel von ScalaSTM (https://nbronson.github.io/scala-stm/) * ... (weiter siehe Digicampus) | | |

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

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

| | | |
|--|---|---|
| Modul INF-0189: Qualitätssicherung im Software Engineering | | 5 ECTS/LP |
| Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| Lernziele/Kompetenzen: Nach erfolgreicher Absolvierung der Lehrveranstaltung verstehen Studierende die Grundtechniken der Qualitätssicherung. Sie sind in der Lage, SW-Module zu spezifizieren und kennen die wesentlichen Testtechniken und deren Einsatzzwecke im Software Engineering. Die Studierenden sind für das Thema Qualität im Software Engineering sensibilisiert und können verschiedene Qualitätskriterien/-metriken kritisch hinterfragen und bewerten. Des Weiteren kennen und verstehen sie die Prinzipien von konstruktiven Qualitätssicherungstechniken und -praktiken. | | |
| Schlüsselqualifikationen: Training des logischen Denkens, analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis, Verbesserung der eigenen Softwareentwicklungskompetenz | | |
| 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) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe) | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Qualitätssicherung im Software Engineering (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| 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 (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

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

Modulteile**Modulteil: 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 zu Medical Information Sciences f. Master (Seminar)**

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

Prüfung**Vortrag und schriftliche Ausarbeitung**

Seminar

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

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

Modulteil: Industrierobotik (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Industrierobotik (Übung)

Prüfung

Industrierobotik (mündliche Prüfung)

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 4 Monate

| | | |
|--|---|---|
| Modul INF-0234: Flugrobotik | | 5 ECTS/LP |
| Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| <p>Inhalte:</p> <p>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.</p> <p>In dieser Veranstaltung liegt der Schwerpunkt auf der Entwicklung von Software für fliegende Robotersysteme. Dazu werden Grundlagen in folgenden Bereichen vermittelt:</p> <ul style="list-style-type: none"> • 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 <p>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.</p> | | |
| <p>Lernziele/Kompetenzen:</p> <p>Die Studierenden kennen die Grundlagen fliegender Roboter und sind in der Lage, diese zu programmieren. Sie kennen den Aufbau von Flugrobotern sowie die Funktionsweise oft verwendeter Sensorik. Sie können fachliche Lösungskonzepte in Software umsetzen und dabei Entwurfsalternativen bewerten und anwenden. Sie haben die Fähigkeit zum analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikation: Analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis, Erwerb von Grundlagen zu fliegenden Robotern.</p> | | |
| <p>Bemerkung:</p> <p>"Grundlagen der Robotik" als Vorkenntnis empfohlen</p> | | |
| <p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std.</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>30 Std. Übung (Präsenzstudium)</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>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> | | |
| <p>Voraussetzungen:</p> <p>keine</p> | | |
| <p>Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)</p> | <p>Empfohlenes Fachsemester: ab dem 1.</p> | <p>Minimale Dauer des Moduls: 1 Semester</p> |
| <p>SWS: 4</p> | <p>Wiederholbarkeit: siehe PO des Studiengangs</p> | |
| <p>Moduleile</p> <p>Modulteil: Flugrobotik (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p> | | |

Literatur:

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

Zugeordnete Lehrveranstaltungen:

Flugrobotik (Vorlesung)

Modulteil: Flugrobotik (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Flugrobotik (Übung)

Prüfung

Flugrobotik

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 4 Monate

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

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

Literatur:

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

Modulteil: Software für Industrie 4.0 (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Prüfung Software für Industrie 4.0

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

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

| |
|---|
| Modulteile |
| <p>Modulteil: Kollaborative Robotik (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p> |
| <p>Literatur:</p> <ul style="list-style-type: none"> • 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 |
| <p>Modulteil: Kollaborative Robotik (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 4</p> |

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

| | | |
|---|--|--|
| Modul INF-0066: Organic Computing II | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| Lernziele/Kompetenzen: Fundierte Kenntnisse über das Forschungsgebiet Organic Computing und die Funktionsweise selbstorganisierender Systeme. Verständnis für Probleme beim Entwurf von komplexen vernetzten Systemen und von forschungsorientierten Lösungsansätzen. Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis | | |
| 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) | | |
| 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 | | |
| 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. | | |
| Literatur: <ul style="list-style-type: none"> • Folien • 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 | | |
| Zugeordnete Lehrveranstaltungen: | | |

Organic Computing II (Vorlesung) (Vorlesung)

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

Modulteil: Organic Computing II (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Organic Computing II (Übung) (Übung)

Prüfung

Organic Computing II (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

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

| | | |
|--|--|--|
| Modul INF-0067: Peer-to-Peer und Cloud Computing | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| Lernziele/Kompetenzen: Erwerb fundierter Kenntnisse über Konzepte und Anwendungen von Cloud-Computing bzw. Peer-to-Peer-Systemen als Grundlage komplexer Internet basierter Infrastrukturen. Dazu werden ein Verständnis für Probleme beim Entwurf von komplexen vernetzten Systemen erarbeitet und forschungsorientierte Lösungsansätze vermittelt. | | |
| Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis | | |
| Arbeitsaufwand: Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) | | |
| Voraussetzungen: | | |
| keine | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Peer-to-Peer und Cloud Computing (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| Inhalte: | | |
| <p>Die Vorlesung "Cloud- und Peer-to-Peer-Computing" vermittelt Ansätze zur Organisation von komplexen abstrahierten IT-Infrastrukturen, die dynamisch an wechselnde Nutzungsbedingungen angepasst werden können und Dienste auf verschiedenen Ebenen zur Verfügung stellen, z.B. Rechenkapazität, Datenspeicher, Netzkapazitäten und Softwaredienste. Dazu werden zunächst Anforderungen und Ziele solcher Systeme definiert und diskutiert. Darüber hinaus werden Konzepte aus dem Bereich der Systemarchitekturen und Ansätze aus dem Bereich selbstorganisierender Algorithmen dargestellt und bewertet. In allen Teilen werden Bezüge zu konkreten Anwendungsgebieten gegeben. Die zugehörige Übung bietet die Möglichkeit, die erlernten Ansätze zu vertiefen und beispielhaft anzuwenden.</p> | | |
| Literatur: | | |
| <ul style="list-style-type: none"> • aktuelle wissenschaftliche Paper • Mahlmann und Schindelbauer: 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 | | |
| 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 | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| <p>Lernziele/Kompetenzen: Erwerb weiterführender Kenntnisse über das Forschungsgebiet Betriebssysteme, basierend auf grundlegenden Konzepten der systemnahen Informatik und Betriebssystemen. Dazu wird ein Verständnis für Probleme bei der Entwicklung moderner Betriebssysteme erarbeitet und anhand von Beispielen illustriert. Die erworbenen Kenntnisse werden anhand von praktischen Übungen vertieft.</p> <p>Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis</p> | | |
| <p>Arbeitsaufwand: Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Weiterführende Betriebssystemkonzepte (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| <p>Inhalte: Die Vorlesung "Weiterführende Betriebssystemkonzepte" vermittelt aufbauend auf den grundlegenden Mechanismen, die bereits aus der Vorlesung "Systemnahe Informatik" bekannt sind, Einblicke in die Funktionsweise von modernen Betriebssystemen. Dabei wird der Fokus des theoretischen Teils auf dem Verständnis von Basismechanismen unter anderem aus den Bereichen Scheduling, Memorymanagement und Input/Output stehen. Der praktische Teil konzentriert sich dabei auf die Umsetzung unterschiedlicher Techniken im Labormaßstab sowie die Evaluation der Leistungsfähigkeit dieser implementierten Konzepte. Grundlage der Arbeiten sind dabei aktuelle Betriebssysteme beispielsweise aus dem Umfeld der Linux und Android Systeme.</p> | | |
| <p>Literatur:</p> <ul style="list-style-type: none"> • Folien • Andrew S. Tanenbaum: "Moderne Betriebssysteme", Pearson Studium, ISBN:978-3-8273-7342-7 | | |
| Zugeordnete Lehrveranstaltungen: | | |
| <p>Weiterführende Betriebssystemkonzepte (Vorlesung) (Vorlesung) Die Vorlesung „Weiterführende Betriebssystemkonzepte“ vermittelt weiterführende Ansätze zur Funktionsweise von Betriebssystemen in verschiedenen Anwendungsgebieten. Dazu werden zunächst Anforderungen und Ziele solcher Systeme definiert und diskutiert. In dieser Vorlesung werden vor allem Besonderheiten der Betriebssysteme aus dem Bereich Sensornetze (z.B. TinyOS), mobile Endgeräte (z.B. Android) und klassischer Desktop- und Serverumgebungen (z.B. Linux) betrachtet und verglichen. Dazu werden beispielsweise Aspekte</p> | | |

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

Modulteil: Weiterführende Betriebssystemkonzepte (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Weiterführende Betriebssystemkonzepte (Übung) (Übung)

Prüfung

Weiterführende Betriebssystemkonzepte (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0070: Seminar Organic Computing | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage zur selbstständigen Erarbeitung eines Themas und der geeigneten Präsentation in Schrift und Vortrag, sowie der sachlichen Diskussion über einen Vortrag. Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis | | |
| Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) | | |
| 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 | | |
| Prüfung Vortrag und schriftliche Ausarbeitung Seminar | | |

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

| | | |
|--|---|---|
| Modul INF-0072: Projektmodul Organic Computing | | 10 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| Lernziele/Kompetenzen: Nach der Teilnahme am Projektmodul sind die Studierenden in der Lage, Problemstellungen höherer Komplexität auf dem Gebiet "Organic Computing" zu verstehen und innovative Konzepte, Methoden, Verfahren, Techniken und Technologien für das genannte Gebiet in Forschungsprojekten zu entwickeln. Sie verfügen über die Team- und Kommunikationsfähigkeit, die Fähigkeit zur Literaturrecherche und die wissenschaftliche Methodik, um Problemstellungen auf dem genannten Gebiet zu diskutieren, Zwischenziele zu definieren sowie Zwischenergebnisse und innovative Ideen kritisch zu bewerten, einzuordnen, zu kombinieren und zu präsentieren. | | |
| Schlüsselqualifikationen: Grundsätze guter wissenschaftlicher Praxis, selbstständiges Arbeiten, Erlernen des Arbeitens mit englischsprachiger Fachliteratur, analytisch-methodische Kompetenz | | |
| Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: nach Bedarf | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 1 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Projektmodul Organic Computing Lehrformen: Praktikum Sprache: Deutsch SWS: 1 | | |
| Inhalte: Mitarbeit an aktuellen Forschungsthemen. | | |
| Literatur: In Abhängigkeit vom zu bearbeitenden Thema: <ul style="list-style-type: none"> • Paper • Buch • Handbuch | | |
| Prüfung Vortrag und Abschlussbericht Praktikum | | |

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

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

| | | |
|--|---|---|
| Modul INF-0088: Bayesian Networks | | 5 ECTS/LP |
| Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Rainer Lienhart | | |
| Lernziele/Kompetenzen: The student understands the core principles of Bayesian Networks and can apply them to many real-world problems of all sorts of different domains such as robots, web search, smart agents, automated diagnosis systems, help systems, and medical systems to name a few. Bayesian Networks are one of the most versatile statistical machine learning technique today. The student will understand, apply, analyse, and evaluate problems from the point of view of Bayesian Networks. | | |
| Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken | | |
| 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) | | |
| Voraussetzungen: keine | | |
| 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 | | |
| Inhalte: <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: <ul style="list-style-type: none"> • 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 | | |
| Zugeordnete Lehrveranstaltungen: Bayesian Networks (Vorlesung) | | |

Modulteil: Bayesian Networks (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Bayesian Networks (Übung)

Prüfung

Bayesian Networks (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0092: Multimedia II: Machine Learning and Computer Vision | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart | | |
| Lernziele/Kompetenzen: Teilnehmer dieser Vorlesung beherrschen wichtige Konzepte des maschinellen Lernens, der Datenreduktion, der fortgeschrittenen Bildverarbeitung und des maschinellen Sehens und können diese anwenden. Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken | | |
| Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vorlesung (Präsenzstudium) | | |
| 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 |
| 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) |
| Literatur: Literaturhinweise werden zu Beginn des Semesters bekanntgegeben. |
| Zugeordnete Lehrveranstaltungen: Multimedia II (Vorlesung) This course addresses state-of-the-art computer vision algorithms that let computers see, learn, and understand image and video content. After being taught the required basics in machine learning, students will - accompanied by practical exercises - get to know the most promising techniques. The topics of the course may be summarized as follows: Machine learning Implementation of algorithms using Intel IPP and MKL Image/video processing Media content analysis The learned concepts will be illustrated by successful examples in practice. The accompanying |

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

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

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Multimedia II (Übung)

Prüfung

Multimedia II: Machine Learning and Computer Vision (Klausur)

Klausur / Prüfungsdauer: 120 Minuten

| | | |
|---|---|---|
| Modul INF-0093: Probabilistic Robotics | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart | | |
| Lernziele/Kompetenzen: This course covers the basics of robot perception and robot motion from a probabilistic point of view. The student is able to understand, apply, analyse, and evaluate problems in robotics from the perspective of probabilistic robotics. This is currently the most successful and modern approach in robotics with impressive performance under uncertainty. | | |
| Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken | | |
| 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) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |

| |
|--|
| Modulteile |
| Modulteil: Probabilistic Robotics (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2 |
| Inhalte: <ol style="list-style-type: none"> 1. Introduction to Probabilistic Robotics 2. Recursive State Estimation 3. Gaussian Filters 4. Nonparametric Filters 5. Robot Motion 6. Robot Perception 7. Mobile Robot Localization: Markov and Gaussian 8. Mobile Robot Localization: Grid and Monte Carlo 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 SWS: 2 |

Prüfung

Probabilistic Robotics (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

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

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

Prüfung

Maschinelles Lernen (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0133: Selbstorganisierende, adaptive Systeme | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| <p>Lernziele/Kompetenzen: Die Studierenden kennen die Eigenschaften und den Aufbau selbst-organisierender Systeme aus der Biologie, Soziologie, Physik und anderen Bereichen und der systematischen Modellierung und Konstruktion adaptiver Systeme in der Informatik und können solche Systeme analysieren und selbst entwerfen. Sie kennen Vor- und Nachteile verschiedener Entwurfsalternativen und können sie im Kontext der Problemstellung bewerten. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken und können geeignete Methoden auswählen und anwenden und wissenschaftlich aussagekräftige Bewertungen abgeben.</p> <p>Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis</p> | | |
| <p>Arbeitsaufwand: Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes 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> | | |
| <p>Literatur:</p> <ul style="list-style-type: none"> • 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 | | |

Modulteil: Selbstorganisierende, adaptive Systeme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Selbstorganisierende, adaptive Systeme (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

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

Prüfung

Mikrorechnertechnik und Echtzeitsysteme (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

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

Modulteil: Cyber-Physical Systems (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 1

Prüfung

Cyber-Physical Systems (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

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

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

Prüfung

Prozessorarchitektur (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

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

Modulteil: Entwurf und Analyse fehlertolerierender Rechensysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Entwurf und Analyse fehlertolerierender Rechensysteme (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

| | | |
|---|--|--|
| Modul INF-0149: Praktikum Eingebettete Systeme | | 5 ECTS/LP |
| Version 1.2.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage Projektaufgaben zu einer Themenstellung aus dem Gebiet "Eingebettete Systeme" einzeln oder Team zu planen, nach einem selbst entwickelten fundierten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren. | | |
| Schlüsselqualifikationen: Projektgebundene Arbeit und Zeitmanagement. | | |
| Arbeitsaufwand: Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) | | |
| Voraussetzungen: Kenntnisse in C. Modul Mikrorechner-technik und Echtzeitsysteme (INF-0145) - empfohlen | | |
| Angebotshäufigkeit: jedes Sommersemester | 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 | | |
| 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. | | |
| Literatur: <ul style="list-style-type: none"> • Marwedel, Wehmeyer: Eingebettete Systeme, Springer Verlag, Heidelberg, 2007 • Wiegmann: Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller, 6. Auflage, VDE Verlag, Berlin, 2011 | | |
| Zugeordnete Lehrveranstaltungen: | | |
| Eingebettete Systeme (Praktikum) | | |
| Prüfung | | |
| Erfolgreiche Praktikumsteilnahme, Projektvorstellung Praktikum | | |

| | | |
|---|---|---|
| Modul INF-0150: Hardware-Entwurf | | 8 ECTS/LP |
| Version 1.0.0 (seit WS14/15) Modulverantwortliche/r: Prof. Dr. Theo Ungerer | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage Projektaufgaben zu einer Themenstellung aus dem Gebiet Prozessorarchitektur im Team zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren. | | |
| Schlüsselqualifikation: Projektgebundene Arbeit und Zeitmanagement | | |
| Arbeitsaufwand: Gesamt: 240 Std. 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) | | |
| Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen | | |
| Angebotshäufigkeit: jedes Wintersemester | 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 | | |
| 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 | | |
| Literatur: | | |
| <ul style="list-style-type: none"> • Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010 • John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011 | | |
| Modulteil: Hardware-Entwurf (Übung) | | |
| Lehrformen: Übung | | |
| Sprache: Deutsch | | |
| SWS: 4 | | |

Prüfung

Projektvorstellung und Projektabnahme

Praktikum

Beschreibung:

Erfolgreiche Praktikumsteilnahme, Projektvorstellung am Ende des Semesters

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

| | | |
|--|---|---|
| Modul INF-0152: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer | | |
| <p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet der Prozessorarchitekturen selbstständig zu erarbeiten, zu analysieren und bezogen auf das individuelle Seminarthema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz entsprechender Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren und zu bewerten.</p> <p>Schlüsselqualifikationen: Grundsätze guter wissenschaftlicher Praxis, Zeitmanagement, Literaturrecherche, Arbeit mit englischsprachiger Fachliteratur</p> | | |
| <p>Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Sommersemester | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Moduleile | | |
| Modulteil: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| <p>Inhalte: Im Seminar werden Architekturen und Technologien moderner Prozessoren aus Forschung und Industrie behandelt. Jeder Seminarteilnehmer erhält individuelle Literaturhinweise, die dann im Laufe des Seminars durch weitere eigenständig erarbeitete Referenzen ergänzt werden sollen. Abschluss des Seminars stellt eine schriftliche Ausarbeitung sowie ein Vortrag über das behandelte Thema dar.</p> | | |
| <p>Literatur: individuell gegeben und Selbstrecherche</p> | | |
| <p>Zugeordnete Lehrveranstaltungen: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen (Seminar) Der Schwerpunkt liegt dieses Semester auf neuartigen Technologien und wie sie die Prozessorarchitektur beeinflussen könnten. Mögliche Themen sind Nanotubes, Graphen, 3D-Stacking, Spintronik, Memristoren, Quantencomputer, künstliche neuronale Netze, nichtflüchtige Speicher, ...</p> | | |
| Prüfung | | |
| Vortrag und schriftliche Ausarbeitung Seminar | | |

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

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

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

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

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

Literatur:

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

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

Klausur / Prüfungsdauer: 100 Minuten

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

Prüfung

Computational Intelligence (mündliche Prüfung und Projektabnahme)

Mündliche Prüfung

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

| |
|---|
| 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>Zugeordnete Lehrveranstaltungen: Multimodale Echtzeitsignalverarbeitung (Praktikum) Social Signal Interpretation (SSI) ist ein an unserem Lehrstuhl entwickelte Toolbox zur parallelen und synchronen Verarbeitung von Sensor Daten in Echtzeit. Es hilft bei der Entwicklung multimodaler Applikationen, wobei es die Verbindung zu den angeschlossenen Sensoren herstellt und nötige Vorverarbeitung, wie Filtering oder Featureberechnung, bis hin zum Training und Einsatz von Klassifikatoren ermöglicht. In dem Praktikum werden verschiedene Aufgabenstellungen angeboten, die jeweils in kleinen Gruppen bearbeitet werden und die Funktionalität der Toolbox erweitern sollen. Die nötigen Grundkenntnisse zur Bearbeitung des Praktikums werden zu Beginn des Praktikums vermittelt: dies beinhaltet eine Einführung in digitale Signalverarbeitung, die</p> |

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

Prüfung

Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation

Projektarbeit

| | | |
|---|---|---|
| Modul INF-0216: Vertiefte Multicore-Programmierung | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Theo Ungerer | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage, vertiefte Themenstellungen aus dem Gebiet der parallelen Programmierung von Multicores im Team zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren. Sie lernen verschiedene Paradigmen der parallelen Programmierung einzuschätzen. Schlüsselqualifikationen: Abstraktes Denken, Projektgebundene Arbeit in Teams, Präsentation und Diskussion von Ergebnissen, Zeitmanagement | | |
| Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Sommersemester | 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: <ul style="list-style-type: none"> • 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. Rüniger: Parallele Programmierung, Springer Verlag, 2007, ISBN: 978-3540465492 • es werden die jeweils neuesten APIs/Unterlagen aus dem Internet verwendet | | |
| Zugeordnete Lehrveranstaltungen: Vertiefte Multicore-Programmierung (Vorlesung) In der Vorlesung und Übung werden die folgenden Technologien behandelt: - C++11 Threads - OpenMP - Transaktionaler Speicher (Intel TSX) - Nachrichtengekoppelte Programmierung (Message Passing Interface, MPI) - GPU-Programmierung (CUDA) | | |

Modulteil: Vertiefte Multicore-Programmierung (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Inhalte:

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

Zugeordnete Lehrveranstaltungen:

Übung zu Vertiefte Multicore-Programmierung (Übung)

Prüfung

Vertiefte Multicore-Programmierung (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Beschreibung:

Projektvorstellung und -abnahme, Fragen zu Vorlesung und Übung

| | | |
|---|---|--|
| Modul INF-0251: Seminar Artificial Intelligence | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: PD Dr. Jonghwa Kim | | |
| Inhalte: Das Seminar wird als Blockseminar Ende Juni für SS oder Mitte Dezember für WS stattfinden. Der Themenbereich für dieses Seminar wird jährlich unter Berücksichtigung neuer Trends in "Künstliche Intelligenz und Intelligente Systeme" neu festgelegt. | | |
| Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, grundlegende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "Künstliche Intelligenz und Intelligente Systeme" selbstständig zu erarbeiten und bezogen auf ein spezielles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die Arbeitstechniken, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren. | | |
| Schlüsselqualifikationen: Erlernen von Präsentationstechniken, Literaturrecherche, Arbeit mit englischer Fachliteratur, Grundsätze guter wissenschaftlicher Praxis | | |
| Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium) | | |
| 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 | | |
| Zugeordnete Lehrveranstaltungen: | | |
| Seminar Artificial Intelligence (Seminar) Das Seminar Artificial Intelligence wird jedes Jahr als Blockseminar entweder Ende Juni zum SS oder Mitte Dezember zum WS stattfinden. Der Themenbereich für dieses Seminar wird jährlich unter Berücksichtigung neuer Trends in "Künstliche Intelligenz und Intelligente Systeme" neu festgelegt. | | |

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

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

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

| |
|--|
| Prüfung Automotive Software Engineering (mündl. Prüfung) Mündliche Prüfung / Prüfungsdauer: 30 Minuten |
|--|

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

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

Prüfung

Praktikum Automotive Software Engineering

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

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

Inhalte:

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

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

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

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

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

Die erforderlichen Tätigkeiten sind:

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

Vorkenntnisse:

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

Literatur:

abhängig vom Thema

Zugeordnete Lehrveranstaltungen:

Praktikum Avionic Software Engineering (Praktikum)

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

Prüfung

Praktikum Avionic Software Engineering

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

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

| | | |
|--|---|---|
| Modul INF-0041: Seminar Grundlagen des Software Engineering für Avionic Systems (MA) | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Bernhard Bauer | | |
| <p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet des Avionic Software Engineerings selbstständig zu erarbeiten und bezogen auf ein spezielles wissenschaftlich anspruchsvolles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren.</p> <p>Schlüsselqualifikationen: Erlernen von Präsentationstechniken; Abwägen von Lösungsansätzen</p> | | |
| <p>Arbeitsaufwand: 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 | |

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

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

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

| | | |
|---|--|--|
| Modul INF-0066: Organic Computing II | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| Lernziele/Kompetenzen: Fundierte Kenntnisse über das Forschungsgebiet Organic Computing und die Funktionsweise selbstorganisierender Systeme. Verständnis für Probleme beim Entwurf von komplexen vernetzten Systemen und von forschungsorientierten Lösungsansätzen. Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis | | |
| 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) | | |
| 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 | | |
| 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 naturalogischer Algorithmen dargestellt und bewertet. In allen Teilen werden Bezüge zu konkreten Anwendungsgebieten gegeben. Die zugehörige Übung bietet die Möglichkeit, die erlernten Ansätze zu vertiefen und beispielhaft anzuwenden. | | |
| Literatur: <ul style="list-style-type: none"> • Folien • 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 | | |
| Zugeordnete Lehrveranstaltungen: | | |

Organic Computing II (Vorlesung) (Vorlesung)

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

Modulteil: Organic Computing II (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Organic Computing II (Übung) (Übung)

Prüfung

Organic Computing II (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

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

| | | |
|--|--|--|
| Modul INF-0067: Peer-to-Peer und Cloud Computing | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| <p>Lernziele/Kompetenzen: Erwerb fundierter Kenntnisse über Konzepte und Anwendungen von Cloud-Computing bzw. Peer-to-Peer-Systemen als Grundlage komplexer Internet basierter Infrastrukturen. Dazu werden ein Verständnis für Probleme beim Entwurf von komplexen vernetzten Systemen erarbeitet und forschungsorientierte Lösungsansätze vermittelt.</p> <p>Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis</p> | | |
| <p>Arbeitsaufwand: Gesamt: 150 Std. 30 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Peer-to-Peer und Cloud Computing (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| <p>Inhalte: Die Vorlesung "Cloud- und Peer-to-Peer-Computing" vermittelt Ansätze zur Organisation von komplexen abstrahierten IT-Infrastrukturen, die dynamisch an wechselnde Nutzungsbedingungen angepasst werden können und Dienste auf verschiedenen Ebenen zur Verfügung stellen, z.B. Rechenkapazität, Datenspeicher, Netzkapazitäten und Softwaredienste. Dazu werden zunächst Anforderungen und Ziele solcher Systeme definiert und diskutiert. Darüber hinaus werden Konzepte aus dem Bereich der Systemarchitekturen und Ansätze aus dem Bereich selbstorganisierender Algorithmen dargestellt und bewertet. In allen Teilen werden Bezüge zu konkreten Anwendungsgebieten gegeben. Die zugehörige Übung bietet die Möglichkeit, die erlernten Ansätze zu vertiefen und beispielhaft anzuwenden.</p> | | |
| <p>Literatur:</p> <ul style="list-style-type: none"> • aktuelle wissenschaftliche Paper • Mahlmann und Schindelhauer: Peer-to-Peer Netzwerke - Algorithmen und Methoden, Springer 2007 • Antonopoulos und Gillam: Cloud Computing - Principles, Systems and Applications, Springer 2010 | | |
| Modulteil: Peer-to-Peer und Cloud Computing (Übung) | | |
| Lehrformen: Übung | | |
| Sprache: Deutsch | | |
| 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 | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| <p>Lernziele/Kompetenzen: Erwerb weiterführender Kenntnisse über das Forschungsgebiet Betriebssysteme, basierend auf grundlegenden Konzepten der systemnahen Informatik und Betriebssystemen. Dazu wird ein Verständnis für Probleme bei der Entwicklung moderner Betriebssysteme erarbeitet und anhand von Beispielen illustriert. Die erworbenen Kenntnisse werden anhand von praktischen Übungen vertieft.</p> <p>Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis</p> | | |
| <p>Arbeitsaufwand: Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Weiterführende Betriebssystemkonzepte (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| <p>Inhalte: Die Vorlesung "Weiterführende Betriebssystemkonzepte" vermittelt aufbauend auf den grundlegenden Mechanismen, die bereits aus der Vorlesung "Systemnahe Informatik" bekannt sind, Einblicke in die Funktionsweise von modernen Betriebssystemen. Dabei wird der Fokus des theoretischen Teils auf dem Verständnis von Basismechanismen unter anderem aus den Bereichen Scheduling, Memorymanagement und Input/Output stehen. Der praktische Teil konzentriert sich dabei auf die Umsetzung unterschiedlicher Techniken im Labormaßstab sowie die Evaluation der Leistungsfähigkeit dieser implementierten Konzepte. Grundlage der Arbeiten sind dabei aktuelle Betriebssysteme beispielsweise aus dem Umfeld der Linux und Android Systeme.</p> | | |
| <p>Literatur:</p> <ul style="list-style-type: none"> • Folien • Andrew S. Tanenbaum: "Moderne Betriebssysteme", Pearson Studium, ISBN:978-3-8273-7342-7 | | |
| Zugeordnete Lehrveranstaltungen: | | |
| <p>Weiterführende Betriebssystemkonzepte (Vorlesung) (Vorlesung) Die Vorlesung „Weiterführende Betriebssystemkonzepte“ vermittelt weiterführende Ansätze zur Funktionsweise von Betriebssystemen in verschiedenen Anwendungsgebieten. Dazu werden zunächst Anforderungen und Ziele solcher Systeme definiert und diskutiert. In dieser Vorlesung werden vor allem Besonderheiten der Betriebssysteme aus dem Bereich Sensornetze (z.B. TinyOS), mobile Endgeräte (z.B. Android) und klassischer Desktop- und Serverumgebungen (z.B. Linux) betrachtet und verglichen. Dazu werden beispielsweise Aspekte</p> | | |

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

Modulteil: Weiterführende Betriebssystemkonzepte (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Weiterführende Betriebssystemkonzepte (Übung) (Übung)

Prüfung

Weiterführende Betriebssystemkonzepte (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0070: Seminar Organic Computing | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Jörg Hähner | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage zur selbstständigen Erarbeitung eines Themas und der geeigneten Präsentation in Schrift und Vortrag, sowie der sachlichen Diskussion über einen Vortrag. | | |
| Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis | | |
| Arbeitsaufwand: Gesamt: 120 Std. 30 Std. Seminar (Präsenzstudium) 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) | | |
| 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 | | |
| Prüfung Vortrag und schriftliche Ausarbeitung Seminar | | |

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

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

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

| |
|--|
| Prüfung Vortrag und Abschlussbericht Praktikum |
|--|

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

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

| |
|--|
| Modulteile |
| Modulteil: Projektmodul Kommunikationssysteme Lehrformen: Praktikum Sprache: Deutsch SWS: 1 |
| Inhalte: Aktuelle Forschungsthemen auf dem Gebiet "Kommunikationssysteme". |
| Literatur: wissenschaftliche Papiere, Handbücher |

| |
|--|
| Prüfung Vortrag und Abschlussbericht Praktikum |
|--|

| | | |
|--|---|---|
| Modul INF-0088: Bayesian Networks | | 5 ECTS/LP |
| Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Rainer Lienhart | | |
| Lernziele/Kompetenzen: The student understands the core principles of Bayesian Networks and can apply them to many real-world problems of all sorts of different domains such as robots, web search, smart agents, automated diagnosis systems, help systems, and medical systems to name a few. Bayesian Networks are one of the most versatile statistical machine learning technique today. The student will understand, apply, analyse, and evaluate problems from the point of view of Bayesian Networks. | | |
| Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken | | |
| 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) | | |
| Voraussetzungen: keine | | |
| 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 | | |
| Inhalte: <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: <ul style="list-style-type: none"> • 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 | | |
| Zugeordnete Lehrveranstaltungen: Bayesian Networks (Vorlesung) | | |

Modulteil: Bayesian Networks (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Bayesian Networks (Übung)

Prüfung

Bayesian Networks (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0092: Multimedia II: Machine Learning and Computer Vision | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart | | |
| Lernziele/Kompetenzen: Teilnehmer dieser Vorlesung beherrschen wichtige Konzepte des maschinellen Lernens, der Datenreduktion, der fortgeschrittenen Bildverarbeitung und des maschinellen Sehens und können diese anwenden. Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken | | |
| Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Übung (Präsenzstudium) 60 Std. Vorlesung (Präsenzstudium) | | |
| 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 |
| 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) |
| Literatur: Literaturhinweise werden zu Beginn des Semesters bekanntgegeben. |
| Zugeordnete Lehrveranstaltungen: Multimedia II (Vorlesung) This course addresses state-of-the-art computer vision algorithms that let computers see, learn, and understand image and video content. After being taught the required basics in machine learning, students will - accompanied by practical exercises - get to know the most promising techniques. The topics of the course may be summarized as follows: Machine learning Implementation of algorithms using Intel IPP and MKL Image/video processing Media content analysis The learned concepts will be illustrated by successful examples in practice. The accompanying |

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

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

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Multimedia II (Übung)

Prüfung

Multimedia II: Machine Learning and Computer Vision (Klausur)

Klausur / Prüfungsdauer: 120 Minuten

| | | |
|--|---|---|
| Modul INF-0093: Probabilistic Robotics | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart | | |
| Lernziele/Kompetenzen: This course covers the basics of robot perception and robot motion from a probabilistic point of view. The student is able to understand, apply, analyse, and evaluate problems in robotics from the perspective of probabilistic robotics. This is currently the most successful and modern approach in robotics with impressive performance under uncertainty. | | |
| Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken | | |
| 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) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Probabilistic Robotics (Vorlesung) | | |
| Lehrformen: Vorlesung Sprache: Deutsch SWS: 2 | | |
| Inhalte: <ol style="list-style-type: none"> 1. Introduction to Probabilistic Robotics 2. Recursive State Estimation 3. Gaussian Filters 4. Nonparametric Filters 5. Robot Motion 6. Robot Perception 7. Mobile Robot Localization: Markov and Gaussian 8. Mobile Robot Localization: Grid and Monte Carlo 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 SWS: 2 | | |

Prüfung

Probabilistic Robotics (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|--|---|---|
| Modul INF-0094: Maschinelles Lernen | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Rainer Lienhart | | |
| Lernziele/Kompetenzen: Teilnehmer dieser Veranstaltung verstehen mathematische Grundlagen und Techniken des maschinellen Lernens wie neuronaler Netze und Support Vector Maschinen. Sie können diese analysieren und selbständig auf neue Probleme anwenden. | | |
| Schlüsselqualifikationen: Fertigkeit zum logischen, analytischen und konzeptionellen Denken | | |
| Arbeitsaufwand: Gesamt: 150 Std. 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 30 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: unregelmäßig | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Maschinelles Lernen (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| Inhalte: Maschinelles Lernen wird heutzutage in vielen praktischen Anwendungen benutzt wie in der Roboternavigation, der Klassifizierung von Spam-E-mails oder der Spracherkennung. Maschinelles Lernen steht für das automatische Lernen des Computers aus Erfahrungen bzw. anhand von Beispielen. Es werden hierbei Muster in den Daten erkannt, anhand derer dann verallgemeinert werden kann, um neue, unbekannte Beispiele klassifizieren zu können. In dieser Vorlesung wird eine Einführung in die mathematischen Grundlagen und Techniken des maschinellen Lernens wie beispielsweise Neuronale Netze und Support Vektor Maschinen gegeben, so dass diese verstanden, analysiert und selbständig auf neue Probleme angewendet werden können. Die behandelten Themen umfassen Wahrscheinlichkeitsverteilungen, Lineare Modelle für Regression und Klassifikation, Neuronale Netze, Kernel Methoden, Sparse Kernel Maschinen und das Kombinieren von Modellen. | | |
| Literatur: <ul style="list-style-type: none"> • Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, Berlin, ISBN-13: 978-0387310732 | | |
| Modulteil: Maschinelles Lernen (Übung) | | |
| Lehrformen: Übung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |

Prüfung

Maschinelles Lernen (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

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

Inhalte:

Agile Softwareentwicklung:

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

Refactoring

- Code Smells
- Prinzipien des objektorientierten Designs
- Wichtige Refactorings

Testen

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

Requirements Engineering

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

Literatur:

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

Modulteil: Softwaretechnik II (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Softwaretechnik II Klausur

Klausur / Prüfungsdauer: 90 Minuten

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

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

Modulteil: Software- und Systemsicherheit (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Software- und Systemsicherheit (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

| | | |
|---|---|---|
| Modul INF-0133: Selbstorganisierende, adaptive Systeme | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe14) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| <p>Lernziele/Kompetenzen: Die Studierenden kennen die Eigenschaften und den Aufbau selbst-organisierender Systeme aus der Biologie, Soziologie, Physik und anderen Bereichen und der systematischen Modellierung und Konstruktion adaptiver Systeme in der Informatik und können solche Systeme analysieren und selbst entwerfen. Sie kennen Vor- und Nachteile verschiedener Entwurfsalternativen und können sie im Kontext der Problemstellung bewerten. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken und können geeignete Methoden auswählen und anwenden und wissenschaftlich aussagekräftige Bewertungen abgeben.</p> <p>Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis</p> | | |
| <p>Arbeitsaufwand: Gesamt: 240 Std. 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes 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> | | |
| <p>Literatur:</p> <ul style="list-style-type: none"> • 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 | | |

Modulteil: Selbstorganisierende, adaptive Systeme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Prüfung

Selbstorganisierende, adaptive Systeme (mündliche Prüfung)

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

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

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

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

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

Prüfung

Mikrorechnertechnik und Echtzeitsysteme (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

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

Modulteil: Cyber-Physical Systems (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 1

Prüfung

Cyber-Physical Systems (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

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

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

Prüfung

Prozessorarchitektur (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

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

Modulteil: Entwurf und Analyse fehlertolerierender Rechensysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Entwurf und Analyse fehlertolerierender Rechensysteme (Klausur)

Klausur / Prüfungsdauer: 60 Minuten

| | | |
|---|--|--|
| Modul INF-0149: Praktikum Eingebettete Systeme | | 5 ECTS/LP |
| Version 1.2.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage Projektaufgaben zu einer Themenstellung aus dem Gebiet "Eingebettete Systeme" einzeln oder Team zu planen, nach einem selbst entwickelten fundierten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren. | | |
| Schlüsselqualifikationen: Projektgebundene Arbeit und Zeitmanagement. | | |
| Arbeitsaufwand: Gesamt: 150 Std. 60 Std. Praktikum (Präsenzstudium) 90 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) | | |
| Voraussetzungen: Kenntnisse in C. Modul Mikrorechner-technik und Echtzeitsysteme (INF-0145) - empfohlen | | |
| Angebotshäufigkeit: jedes Sommersemester | 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 |
| 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. |
| Literatur: <ul style="list-style-type: none"> • Marwedel, Wehmeyer: Eingebettete Systeme, Springer Verlag, Heidelberg, 2007 • Wiegmann: Softwareentwicklung in C für Mikroprozessoren und Mikrocontroller, 6. Auflage, VDE Verlag, Berlin, 2011 |
| Zugeordnete Lehrveranstaltungen: Eingebettete Systeme (Praktikum) |

| |
|--|
| Prüfung Erfolgreiche Praktikumsteilnahme, Projektvorstellung Praktikum |
|--|

| | | |
|--|---|---|
| Modul INF-0150: Hardware-Entwurf | | 8 ECTS/LP |
| Version 1.0.0 (seit WS14/15) Modulverantwortliche/r: Prof. Dr. Theo Ungerer | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage Projektaufgaben zu einer Themenstellung aus dem Gebiet Prozessorarchitektur im Team zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren. | | |
| Schlüsselqualifikation: Projektgebundene Arbeit und Zeitmanagement | | |
| Arbeitsaufwand: Gesamt: 240 Std. 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) | | |
| Voraussetzungen: Modul Systemnahe Informatik (INF-0138) - empfohlen | | |
| Angebotshäufigkeit: jedes Wintersemester | 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 |
| 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 |
| Literatur: <ul style="list-style-type: none"> • Uwe Brinkschulte, Theo Ungerer, Mikrocontroller und Mikroprozessoren, Springer Verlag, Heidelberg, dritte Auflage 2010 • John L. Hennessy, David A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, 5. Auflage, 2011 |
| Modulteil: Hardware-Entwurf (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 4 |

Prüfung

Projektvorstellung und Projektabnahme

Praktikum

Beschreibung:

Erfolgreiche Praktikumsteilnahme, Projektvorstellung am Ende des Semesters

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

| | | |
|--|---|---|
| Modul INF-0152: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Theo Ungerer | | |
| <p>Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet der Prozessorarchitekturen selbstständig zu erarbeiten, zu analysieren und bezogen auf das individuelle Seminarthema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz entsprechender Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren und zu bewerten.</p> <p>Schlüsselqualifikationen: Grundsätze guter wissenschaftlicher Praxis, Zeitmanagement, Literaturrecherche, Arbeit mit englischsprachiger Fachliteratur</p> | | |
| <p>Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Sommersemester | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Moduleile | | |
| Modulteil: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen | | |
| Lehrformen: Seminar | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| <p>Inhalte: Im Seminar werden Architekturen und Technologien moderner Prozessoren aus Forschung und Industrie behandelt. Jeder Seminarteilnehmer erhält individuelle Literaturhinweise, die dann im Laufe des Seminars durch weitere eigenständig erarbeitete Referenzen ergänzt werden sollen. Abschluss des Seminars stellt eine schriftliche Ausarbeitung sowie ein Vortrag über das behandelte Thema dar.</p> | | |
| <p>Literatur: individuell gegeben und Selbstrecherche</p> | | |
| <p>Zugeordnete Lehrveranstaltungen: Seminar Prozessorarchitekturen: Aktuelle Forschungsthemen (Seminar) Der Schwerpunkt liegt dieses Semester auf neuartigen Technologien und wie sie die Prozessorarchitektur beeinflussen könnten. Mögliche Themen sind Nanotubes, Graphen, 3D-Stacking, Spintronik, Memristoren, Quantencomputer, künstliche neuronale Netze, nichtflüchtige Speicher, ...</p> | | |
| Prüfung | | |
| Vortrag und schriftliche Ausarbeitung Seminar | | |

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

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

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

| |
|---|
| Modulteile |
| Modulteil: Digital Signal Processing II (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 4 |
| Inhalte: Ziel des Moduls ist es, die in der Vorlesung "Digital Signal Processing I" gewonnenen Grundkenntnisse digitaler Signalverarbeitung zu erweitern. Die Vorlesung beginnt mit Zusammenfassung des in der Vorlesung Digital Signal Processing I behandelten Stoffes und bietet eine erweiterte Einführung in folgende Themenbereiche: z-Transformation, Systemfunktion, FIR-/IIR-Filter, Wavelet-Transformation, Subband Coding, Signalverarbeitung für Mustererkennung und Multimedia-Anwendungen. Die Vorlesung wird ergänzt durch integrierte MATLAB-Übungen. |
| Literatur: <ul style="list-style-type: none"> • Alan V. Oppenheim and Roland W. Schaffer, "Discrete-Time Signal Processing", Prentice Hall • K. Mitra, "Digital Signal Processing: A Computer-Based Approach", McGraw-Hill • Stéphane Mallat, "A Wavelet Tour of Signal Processing", Academic Press |

| |
|---|
| Prüfung Digital Signal Processing II (Klausur) Klausur / Prüfungsdauer: 100 Minuten |
|---|

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

Prüfung

Computational Intelligence (mündliche Prüfung und Projektabnahme)

Mündliche Prüfung

| | | |
|--|--|--|
| Modul INF-0182: Praktikum Multimodale Echtzeitsignalverarbeitung | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe13) Modulverantwortliche/r: Prof. Dr. Elisabeth André | | |
| Lernziele/Kompetenzen: Die Studenten sind mit Methoden und Techniken aus dem Gebiet "Multimodale Echtzeitsignalverarbeitung" vertraut. Sie sind in der Lage, in kleinen Teams größere Projektaufgaben (Entwicklung von Softwaremodulen) zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und als Team zu präsentieren. | | |
| Schlüsselqualifikationen: Fertigkeit der sicheren und überzeugenden Darstellung von Ideen und Konzepten; Kenntnisse der Denkweise und Sprache anwendungsrelevanter Disziplinen; Verstehen von Teamprozessen; Fertigkeit der Zusammenarbeit in Teams; Fähigkeit zur Leitung von Teams; Fertigkeit zur verständlichen Darstellung und Dokumentation von Ergebnissen; Fähigkeit, vorhandenes Wissen selbstständig zu erweitern; Fähigkeit, Beiträge zur Wissenschaft zu leisten; Kompetenz zum Erkennen von bedeutenden technischen Entwicklungen; Qualitätsbewusstsein, Akribie | | |
| Arbeitsaufwand: Gesamt: 240 Std. 150 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 90 Std. Praktikum (Präsenzstudium) | | |
| Voraussetzungen: Programmiererfahrung Modul Multimedia Grundlagen I (INF-0087) - empfohlen Modul Multimedia Grundlagen II (INF-0166) - empfohlen | | |
| Angebotshäufigkeit: jedes Sommersemester | 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. |
| Zugeordnete Lehrveranstaltungen: Multimodale Echtzeitsignalverarbeitung (Praktikum) Social Signal Interpretation (SSI) ist ein an unserem Lehrstuhl entwickelte Toolbox zur parallelen und synchronen Verarbeitung von Sensor Daten in Echtzeit. Es hilft bei der Entwicklung multimodaler Applikationen, wobei es die Verbindung zu den angeschlossenen Sensoren herstellt und nötige Vorverarbeitung, wie Filtering oder Featureberechnung, bis hin zum Training und Einsatz von Klassifikatoren ermöglicht. In dem Praktikum werden verschiedene Aufgabenstellungen angeboten, die jeweils in kleinen Gruppen bearbeitet werden und die Funktionalität der Toolbox erweitern sollen. Die nötigen Grundkenntnisse zur Bearbeitung des Praktikums werden zu Beginn des Praktikums vermittelt: dies beinhaltet eine Einführung in digitale Signalverarbeitung, die |

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

Prüfung

Vortrag mit Softwarerepräsentation, Ausarbeitung mit Softwaredokumentation

Projektarbeit

| | | |
|--|---|---|
| Modul INF-0189: Qualitätssicherung im Software Engineering | | 5 ECTS/LP |
| Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| Lernziele/Kompetenzen: Nach erfolgreicher Absolvierung der Lehrveranstaltung verstehen Studierende die Grundtechniken der Qualitätssicherung. Sie sind in der Lage, SW-Module zu spezifizieren und kennen die wesentlichen Testtechniken und deren Einsatzzwecke im Software Engineering. Die Studierenden sind für das Thema Qualität im Software Engineering sensibilisiert und können verschiedene Qualitätskriterien/-metriken kritisch hinterfragen und bewerten. Des Weiteren kennen und verstehen sie die Prinzipien von konstruktiven Qualitätssicherungstechniken und -praktiken. | | |
| Schlüsselqualifikationen: Training des logischen Denkens, analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis, Verbesserung der eigenen Softwareentwicklungskompetenz | | |
| 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) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe) | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Qualitätssicherung im Software Engineering (Vorlesung) | | |
| Lehrformen: Vorlesung | | |
| Sprache: Deutsch | | |
| SWS: 2 | | |
| 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 (Klausur)

Klausur / Prüfungsdauer: 90 Minuten

| | | |
|---|---|---|
| Modul INF-0216: Vertiefte Multicore-Programmierung | | 8 ECTS/LP |
| Version 1.0.0 (seit SoSe16) Modulverantwortliche/r: Prof. Dr. Theo Ungerer | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage, vertiefte Themenstellungen aus dem Gebiet der parallelen Programmierung von Multicores im Team zu planen, nach einem selbst entwickelten Projektplan zu lösen und die Resultate angemessen im Plenum zu diskutieren und zu präsentieren. Sie lernen verschiedene Paradigmen der parallelen Programmierung einzuschätzen. Schlüsselqualifikationen: Abstraktes Denken, Projektgebundene Arbeit in Teams, Präsentation und Diskussion von Ergebnissen, Zeitmanagement | | |
| Arbeitsaufwand: Gesamt: 240 Std. 30 Std. Vorlesung (Präsenzstudium) 60 Std. Übung (Präsenzstudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Sommersemester | 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: <ul style="list-style-type: none"> • 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. Rüniger: Parallele Programmierung, Springer Verlag, 2007, ISBN: 978-3540465492 • es werden die jeweils neuesten APIs/Unterlagen aus dem Internet verwendet | | |
| Zugeordnete Lehrveranstaltungen: Vertiefte Multicore-Programmierung (Vorlesung) In der Vorlesung und Übung werden die folgenden Technologien behandelt: - C++11 Threads - OpenMP - Transaktionaler Speicher (Intel TSX) - Nachrichtengekoppelte Programmierung (Message Passing Interface, MPI) - GPU-Programmierung (CUDA) | | |

Modulteil: Vertiefte Multicore-Programmierung (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Inhalte:

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

Zugeordnete Lehrveranstaltungen:

Übung zu Vertiefte Multicore-Programmierung (Übung)

Prüfung

Vertiefte Multicore-Programmierung (mündliche Prüfung)

Mündliche Prüfung, Dauer: 30-45 Minuten

Beschreibung:

Projektvorstellung und -abnahme, Fragen zu Vorlesung und Übung

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

| |
|--|
| Modulteile |
| Modulteil: 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 zu Medical Information Sciences f. Master (Seminar) Bestandteil dieses Seminars sind fortgeschrittene Ansätze und Techniken im Bereich Medical Information Science. |

| |
|---|
| Prüfung Vortrag und schriftliche Ausarbeitung Seminar |
|---|

| | | |
|--|---|---|
| Modul INF-0233: Industrierobotik | | 8 ECTS/LP |
| Version 2.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| Lernziele/Kompetenzen: Die Studierenden sind in der Lage Industrieroboter zu programmieren. Sie können fachliche Lösungskonzepte in Roboterprogramme umsetzen und dabei Entwurfsalternativen bewerten und anwenden. Sie haben Kenntnisse von praxisrelevanten Aufgabenstellungen in der automatisierten Fertigung. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken. | | |
| Schlüsselqualifikationen: analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis | | |
| 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. 60 Std. Übung (Präsenzstudium) 30 Std. Vorlesung (Präsenzstudium) 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) | | |
| Voraussetzungen: | | |
| keine | | |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 6 | Wiederholbarkeit: siehe PO des Studiengangs | |

| |
|---|
| Modulteile |
| Modulteil: Industrierobotik (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2 |
| Inhalte: Ziel der Veranstaltung ist es, an Beispielen die Programmierung und der Entwurf von Software für Industrieroboter, wie sie z.B. in der Automobilindustrie verwendet werden, zu erlernen. Dazu werden im ersten Teil der Vorlesung Grundlagen wie Kinematik und Bahnplanung mit Hilfe simulationsbasierter Programmieransätze behandelt. Im zweiten Teil der Vorlesung werden verschiedene, kleine Programmieraufgabenstellungen bearbeitet und auf einem KUKA Roboter evaluiert. Die Programmierung erfolgt mit der Roboterprogrammiersprache KRL. |
| Literatur: <ul style="list-style-type: none"> • L. Sciavicco, B. Siciliano: Modelling and Control of Robot Manipulators. Reihe: Advanced Textbooks in Control and Signal Processing. Springer 2000 (2nd Ed.) • Handbücher von KUKA • Folienhandout |
| Zugeordnete Lehrveranstaltungen: Industrierobotik (Vorlesung) Diese Veranstaltung steht unter dem Motto "Uni goes Industry", da in dieser Veranstaltung die Programmierung "echter" Industrieroboter incl. zugehöriger Software-Entwicklungsumgebung und Simulationsmöglichkeiten |

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

Modulteil: Industrierobotik (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 4

Zugeordnete Lehrveranstaltungen:

Übung zu Industrierobotik (Übung)

Prüfung

Industrierobotik (mündliche Prüfung)

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 4 Monate

| | | |
|--|---|---|
| Modul INF-0234: Flugrobotik | | 5 ECTS/LP |
| Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | | |
| <p>Inhalte:</p> <p>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.</p> <p>In dieser Veranstaltung liegt der Schwerpunkt auf der Entwicklung von Software für fliegende Robotersysteme. Dazu werden Grundlagen in folgenden Bereichen vermittelt:</p> <ul style="list-style-type: none"> • 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 <p>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.</p> | | |
| <p>Lernziele/Kompetenzen:</p> <p>Die Studierenden kennen die Grundlagen fliegender Roboter und sind in der Lage, diese zu programmieren. Sie kennen den Aufbau von Flugrobotern sowie die Funktionsweise oft verwendeter Sensorik. Sie können fachliche Lösungskonzepte in Software umsetzen und dabei Entwurfsalternativen bewerten und anwenden. Sie haben die Fähigkeit zum analytischen und konzeptionellen Denken.</p> <p>Schlüsselqualifikation: Analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis, Erwerb von Grundlagen zu fliegenden Robotern.</p> | | |
| <p>Bemerkung:</p> <p>"Grundlagen der Robotik" als Vorkenntnis empfohlen</p> | | |
| <p>Arbeitsaufwand:</p> <p>Gesamt: 150 Std.</p> <p>30 Std. Vorlesung (Präsenzstudium)</p> <p>30 Std. Übung (Präsenzstudium)</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>15 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium)</p> | | |
| <p>Voraussetzungen:</p> <p>keine</p> | | |
| <p>Angebotshäufigkeit: unregelmäßig (i. d. R. im SoSe)</p> | <p>Empfohlenes Fachsemester: ab dem 1.</p> | <p>Minimale Dauer des Moduls: 1 Semester</p> |
| <p>SWS: 4</p> | <p>Wiederholbarkeit: siehe PO des Studiengangs</p> | |
| <p>Moduleile</p> <p>Modulteil: Flugrobotik (Vorlesung)</p> <p>Lehrformen: Vorlesung</p> <p>Sprache: Deutsch</p> <p>SWS: 2</p> | | |

Literatur:

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

Zugeordnete Lehrveranstaltungen:

Flugrobotik (Vorlesung)

Modulteil: Flugrobotik (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Zugeordnete Lehrveranstaltungen:

Übung zu Flugrobotik (Übung)

Prüfung

Flugrobotik

praktische Prüfung / Prüfungsdauer: 45 Minuten

Bearbeitungsfrist: 4 Monate

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

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

Literatur:

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

Modulteil: Software für Industrie 4.0 (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Prüfung Software für Industrie 4.0

Mündliche Prüfung / Prüfungsdauer: 45 Minuten

| | |
|--|-----------|
| Modul INF-0248: Kollaborative Robotik | 8 ECTS/LP |
| Version 1.0.0 Modulverantwortliche/r: Prof. Dr. Wolfgang Reif | |
| <p>Inhalte:</p> <p>Die Digitalisierung dringt mittlerweile in viele Bereiche des täglichen Lebens vor. Von Industriemaschinen und Robotersystemen (<i>Smart Factory, Industrie 4.0</i>) bis hin zu intelligenten Hausgeräten oder Heizungsanlagen (<i>Smart Home</i>) findet eine immer stärkere Vernetzung dieser Geräte im Internet der Dinge statt. Hierbei spricht man von cyber-physischen Systemen, die einerseits durch ein komplexes Zusammenspiel von vernetzten eingebetteten Systemen entstehen und andererseits geprägt sein werden von einer vollkommen neuen Art der Mensch-Technik-Interaktion in den Anwendungen.</p> <p>Eine besondere Rolle unter den cyber-physischen Systemen nehmen Assistenz- oder Serviceroboter ein, da sie durch einen geteilten Arbeitsraum in eine direkte, auch physische Interaktion mit dem Menschen treten können. Diese robotischen Systeme können den Menschen in seiner täglichen Arbeit unterstützen bzw. entlasten. Das gilt sowohl im häuslichen Umfeld als auch in der Fabrik. Im Kontext von Industrie 4.0 haben robotische Co-Worker (CoBots) vereinzelt bereits Eingang in die Produktion gefunden. Diese kollaborativen Einsatzszenarien werden jedoch in Zukunft noch weiter an Bedeutung gewinnen. Durch die in 2016 veröffentlichte Norm ISO/TS 15066:2016 zur Mensch-Roboter-Kollaboration stehen dem Einsatz innovativer Robotersysteme neue Möglichkeiten offen.</p> <p>Während die grundlegende Hardware, d.h. die Roboter, mobilen Plattformen, Greifer und Sensoren, bereitgestellt wird, liegt der Schwerpunkt der Veranstaltung auf der Entwicklung von Software für kollaborative (mobile) Manipulatoren. Durch die intelligente Vernetzung von Aktuatorik und Sensorik, neuartige Algorithmen für die Sensorauswertung und kreative Konzepte zur (physischen) Interaktion mit dem Menschen entstehen innovative, kollaborative Robotersysteme im praktischen Teil der Veranstaltung.</p> <p>Infolgedessen liegt der thematische Fokus der Veranstaltung auf</p> <ul style="list-style-type: none"> • 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. <p>Diese Kernelemente bieten die Grundlage, um eine praktische Aufgabenstellung prototypisch umzusetzen. Die Programmierung erfolgt dabei in objektorientierten Programmiersprachen. So können kollaborative Roboter bspw. in Java mit der Robotics API programmiert werden, einem am ISSE in Kooperation mit KUKA entwickelten Roboterframework. Alternativ oder ergänzend kann das Robot Operating System (ROS) verwendet werden, das in den Programmiersprachen C++ und Python programmiert werden kann.</p> | |
| <p>Lernziele/Kompetenzen:</p> <p>Die Studierenden sind in Lage kollaborative Roboter zu programmieren und kennen die regulatorischen Randbedingungen (vgl. ISO/TS 15066:2016). Sie kennen die grundlegenden Konzepte und Methoden der kollaborativen Robotik. Sie können dementsprechend fachliche Lösungskonzepte umsetzen und Entwurfsalternativen bewerten und anwenden. Sie haben die Fertigkeit zum analytischen und konzeptionellen Denken und können grundlegende Bewertungen der Risiken eines kollaborativen Robotersystems durchführen.</p> <p>Schlüsselqualifikationen:</p> <p>Analytisch-methodische Kompetenz, Abwägen von Lösungsansätzen, Erwerb von Abstraktionsfähigkeiten, Grundsätze guter wissenschaftlicher Praxis, Erwerb von Grundlagen für die Sicherheitsbewertung.</p> | |
| <p>Bemerkung:</p> <p>Voraussetzungen (empfohlen):</p> <p>Software für Industrie 4.0 oder Industrierobotik oder Grundlagen der Robotik</p> | |
| <p>Arbeitsaufwand:</p> <p>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 (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 2</p> |
| <p>Literatur:</p> <ul style="list-style-type: none"> • 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 |
| <p>Modulteil: Kollaborative Robotik (Übung) Lehrformen: Übung Sprache: Deutsch SWS: 4</p> |

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

| | | |
|---|---|--|
| Modul INF-0251: Seminar Artificial Intelligence | | 4 ECTS/LP |
| Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: PD Dr. Jonghwa Kim | | |
| Inhalte: Das Seminar wird als Blockseminar Ende Juni für SS oder Mitte Dezember für WS stattfinden. Der Themenbereich für dieses Seminar wird jährlich unter Berücksichtigung neuer Trends in "Künstliche Intelligenz und Intelligente Systeme" neu festgelegt. | | |
| Lernziele/Kompetenzen: Nach dem Besuch des Seminars sind die Studierenden in der Lage, grundlegende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet "Künstliche Intelligenz und Intelligente Systeme" selbstständig zu erarbeiten und bezogen auf ein spezielles Thema aus dem genannten Gebiet zu bewerten. Sie verfügen über die Arbeitstechniken, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren. | | |
| Schlüsselqualifikationen: Erlernen von Präsentationstechniken, Literaturrecherche, Arbeit mit englischer Fachliteratur, Grundsätze guter wissenschaftlicher Praxis | | |
| Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium) | | |
| 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 | | |
| Zugeordnete Lehrveranstaltungen: | | |
| Seminar Artificial Intelligence (Seminar) Das Seminar Artificial Intelligence wird jedes Jahr als Blockseminar entweder Ende Juni zum SS oder Mitte Dezember zum WS stattfinden. Der Themenbereich für dieses Seminar wird jährlich unter Berücksichtigung neuer Trends in "Künstliche Intelligenz und Intelligente Systeme" neu festgelegt. | | |

Prüfung

Vortrag und schriftliche Ausarbeitung

Seminar

| | | |
|---|--|--|
| Modul INF-0236: Digitale Regelsysteme <i>Digital Control Systems (in German language)</i> | | 6 ECTS/LP |
| Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Christoph Ament | | |
| Lernziele/Kompetenzen: Kostengünstige Sensoren und ebenfalls sehr preiswert verfügbare Rechenleistung erlauben es heute, Prozesse umfassend zu erfassen bzw. aufwändige 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? Die Vorlesung vermittelt Ihnen Werkzeuge, für den Entwurf dieses "digitalen Regelsystems". Sie kennen somit den Aufbau eines digitalen Regelsystems und können dieses auf eine Projektaufgabe übertragen. Sie können geeignete modellbasierte Entwurfsverfahren anwenden, um eine entsprechende Software zu entwickeln und Ihre Diagnose- oder Regelungsaufgabe zu lösen. | | |
| Arbeitsaufwand: Gesamt: 180 Std. 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 23 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 22 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 45 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) | | |
| Voraussetzungen: Als Voraussetzung für diese Veranstaltung sind 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. | | |
| Angebotshäufigkeit: jedes Wintersemester | Empfohlenes Fachsemester: 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 5 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Digitale Regelsysteme (Vorlesung) Lehrformen: Vorlesung Sprache: Deutsch SWS: 3 | | |

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 das Kalman-Filter zur Schätzung dynamischer Zustände entwickelt.

Gliederung:

1. Einführung: Ziele und Aufbau eines digitalen Regelsystems

Teil A: Zeitdiskrete Systeme

2. Beschreibung mittels z-Transformation
3. Beschreibung im Zustandsraum
4. Analyse von Systemeigenschaften

Teil B: Modellbasierter Reglerentwurf

5. Rapid Control Prototyping
6. Reglerentwurfsverfahren für lineare Systeme (Eigenwertvorgabe, Entwurf auf Endliche Einstellzeit)
7. Reglerentwurfsverfahren für nichtlineare Systeme (Modell Predictive Control)

Teil C: Modellbasierte Diagnose

8. Parameterschätzung
9. Kalman-Filter

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.

Modulteil: Digitale Regelsysteme (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Prüfung

Digitale Regelsysteme

Schriftlich-Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

Die Wiederholungsprüfung findet nach der Vorlesungszeit des Sommersemesters statt.

| | | |
|--|---|---|
| Modul INF-0237: Projektmodul Regelungstechnik | | 10 ECTS/LP |
| Version 1.0.0 (seit WS16/17) Modulverantwortliche/r: Prof. Dr. Christoph Ament | | |
| Lernziele/Kompetenzen: Die Studierenden haben tiefere Fachkenntnisse und Fähigkeiten auf dem Gebiet der Regelungstechnik erworben, die es ihnen ermöglichen, an die internationale Forschung anzuknüpfen. Sie sind fähig, innovative Methoden bei der Lösung von Problemen in diesem Gebiet anzuwenden und einen wissenschaftlichen Beitrag zu diesem Gebiet zu leisten. | | |
| Schlüsselqualifikationen: Grundsätze guter wissenschaftlicher Praxis, selbstständiges Arbeiten, Erlernen des Arbeitens mit englischsprachiger Fachliteratur, analytisch-methodische Kompetenz | | |
| Arbeitsaufwand: Gesamt: 300 Std. 285 Std. Praktikum (Selbststudium) 15 Std. Seminar (Präsenzstudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: nach Bedarf | Empfohlenes Fachsemester: ab dem 1. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 1 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Projektmodul Regelungstechnik | | |
| Lehrformen: Praktikum | | |
| Sprache: Deutsch | | |
| Inhalte: Mitarbeit an aktuellen Forschungsthemen des Lehrstuhls | | |
| Literatur: abhängig von dem konkreten Projekt: wissenschaftliche Papiere, Dokumentation | | |
| Prüfung | | |
| Projektabschluss: Praktikum | | |

| | | |
|--|---|---|
| Modul INF-0238: Digitale Fabrik | | 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 grundlegenden Konzepte der Digitalen Fabrik und ihre Anwendung zu vermitteln. Die Studenten kennen die Einsatzmöglichkeiten von Werkzeugen der Digitalen Fabrik in produzierenden Unternehmen und können die Potentiale im Kontext konkreter Fragestellungen bewerten.</p> <p>Schlüsselqualifikation: Analytisch-methodische Kompetenz, Abstraktionsfähigkeit, anwendungsorientierte Problemlösung</p> | | |
| <p>Arbeitsaufwand: Gesamt: 180 Std. 45 Std. Vorlesung (Präsenzstudium) 30 Std. Übung (Präsenzstudium) 15 Std. Vor- und Nachbereitung des Stoffes durch Literatur (Selbststudium) 30 Std. Vor- und Nachbereitung des Stoffes anhand bereitgestellter Unterlagen (Selbststudium) 60 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium)</p> | | |
| <p>Voraussetzungen: Modul Prozessmodellierung und Produktionssteuerung (INF-0197) - empfohlen</p> | | |
| <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> | | |
| <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:</p> <ul style="list-style-type: none"> · 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 | | |
| <p>Literatur: wird in der Vorlesung bekannt gegeben</p> | | |

Modulteil: Digitale Fabrik (Übung)

Lehrformen: Übung

Sprache: Deutsch

SWS: 2

Inhalte:

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

Prüfung

Digitale Fabrik

Schriftlich-Mündliche Prüfung / Prüfungsdauer: 30 Minuten

Beschreibung:

Die Wiederholungsprüfung findet nach der Vorlesungszeit des Sommersemesters statt.

| | | |
|---|---|---|
| Modul INF-0244: Projektmodul Produktionsinformatik | | 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: Im Rahmen des Projektmoduls erwerben die Studierenden tiefere Fachkenntnisse und Fähigkeiten, die es ihnen ermöglichen, an die internationale Forschung anzuknüpfen. Sie sind fähig, innovative Methoden bei der Lösung von Problemen im betrachteten Themengebiet anzuwenden und einen wissenschaftlichen Beitrag zu leisten. | | |
| Schlüsselqualifikationen: Grundsätze guter wissenschaftlicher Praxis, selbstständiges Arbeiten, Erlernen des Arbeitens mit englischsprachiger Fachliteratur, Literaturrecherche, analytisch-methodische Kompetenz, Ergebnisbewertung | | |
| Bemerkung: Ersetzt "Projektmodul Industrie 4.0" (INF-0239) | | |
| Arbeitsaufwand: Gesamt: 300 Std. 15 Std. Seminar (Präsenzstudium) 285 Std. Praktikum (Selbststudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: nach Bedarf | 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 | | |
| Prüfung | | |
| Projektabnahme Praktikum | | |

| | | |
|---|---|---|
| Modul INF-0245: Intelligent vernetzte Produktion | | 5 ECTS/LP |
| Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr.-Ing. Johannes Schilp | | |
| <p>Inhalte: Im Zuge der Vorlesung Intelligent vernetzte Produktion werden den Studierenden folgende Ebenen der Vernetzung in der industriellen Produktion vermittelt:</p> <ul style="list-style-type: none"> - 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 <p>Technologien sowie mögliche Ausprägungen und Strategien zur Vernetzung in den jeweiligen Bereichen werden besprochen.</p> <p>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.</p> | | |
| <p>Lernziele/Kompetenzen: Ziel der Vorlesung Intelligent vernetzte Produktion ist es, den Studierenden vertieftes Verständnis über Vernetzungen und resultierende Optimierungsmöglichkeiten in der Produktion zu vermitteln. Die Studierenden sind in der Lage, Vernetzungen auf Mikro- und Makroebene im Umfeld industrieller Produktion zu analysieren und erlernte Methoden zur Optimierung anzuwenden.</p> <p>Schlüsselqualifikationen: Analytisch-methodische Kompetenz, Abstraktionsfähigkeit, anwendungsorientierte Problemlösung, Abwägen von Lösungsansätzen.</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: Modul Prozessmodellierung und Produktionssteuerung (INF-0197) empfohlen | | |
| 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> | | |
| <p>Zugeordnete Lehrveranstaltungen: Intelligent vernetzte Produktion (Vorlesung)</p> | | |

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.

Zugeordnete Lehrveranstaltungen:

Übung zu Intelligent vernetzte Produktion (Übung)

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 | | 4 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 dem Besuch des Seminars sind die Studierenden in der Lage, weiterführende Problemstellungen, Konzepte, Methoden, Verfahren, Techniken und Technologien auf dem Gebiet der Produktionsinformatik selbstständig zu erarbeiten, zu analysieren und bezogen auf das individuelle Seminarthema aus dem genannten Gebiet zu bewerten. Sie verfügen über die wissenschaftliche Methodik, Kommunikationsfähigkeit und Fähigkeit zum Einsatz neuer Medien, um ein spezielles Thema in Wort und Schrift klar und verständlich zu präsentieren und wissenschaftlich anspruchsvolle Themenstellungen aus dem genannten Gebiet kritisch und argumentativ zu diskutieren und zu bewerten. | | |
| Schlüsselqualifikationen: Grundsätze guter wissenschaftlicher Praxis, selbstständiges Arbeiten, Erlernen des Arbeitens mit englischsprachiger Fachliteratur, Literaturrecherche, analytisch-methodische Kompetenz, Ergebnisbewertung. | | |
| Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Sommersemester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Seminar Industrie 4.0 Lehrformen: Seminar Sprache: Deutsch SWS: 2 ECTS/LP: 4.0 | | |
| Zugeordnete Lehrveranstaltungen: Seminar zu Industrie 4.0 (Seminar) | | |
| Prüfung Seminar Industrie 4.0 Seminar | | |

| | | |
|---|---|---|
| Modul INF-0247: Praktikum für Produktionsinformatik (Vertiefung) | | 6 ECTS/LP |
| Version 1.0.0 (seit SoSe17) Modulverantwortliche/r: Prof. Dr.-Ing. Johannes Schilp | | |
| Inhalte: Die Studenten bearbeiten in Kleingruppen anwendungsorientierte Aufgaben zu Themenbereichen im industriellen Umfeld. | | |
| Lernziele/Kompetenzen: Die Studierenden erweitern ihre Kenntnisse zu Anwendungsfeldern und Technologien von Industrie 4.0 im Bereich der industriellen Produktion anhand praxisnaher Anwendungsfälle. Sie vertiefen Konzepte, Methoden, Verfahren und Techniken aus dem Bereich der Produktionsinformatik, die in den Vorlesungen behandelt wurden. | | |
| Schlüsselqualifikationen: Team- und Kommunikationsfähigkeit, selbstständiges Arbeiten, anwendungsorientierte Problemlösung, Ergebnisbewertung. | | |
| Arbeitsaufwand: Gesamt: 180 Std. 120 Std. Vor- und Nachbereitung des Stoffes Übung/Fallstudien (Selbststudium) 60 Std. Praktikum (Präsenzstudium) | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Sommersemester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 4 | Wiederholbarkeit: siehe PO des Studiengangs | |

| |
|---|
| Modulteile |
| Modulteil: Praktikum für Produktionsinformatik (Vertiefung) Lehrformen: Praktikum Sprache: Deutsch SWS: 4 ECTS/LP: 6.0 |
| Zugeordnete Lehrveranstaltungen: Praktikum für Produktionsinformatik (Vertiefung) (Praktikum) |

| |
|--|
| 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: Die Teilnehmerinnen und Teilnehmer können Literatur zu einem ingenieurwissenschaftlichen Thema eigenständig recherchieren, inhaltlich bearbeiten, bewerten und in den Kontext einer Forschungsfrage einordnen. Schlüsselqualifikationen: Eigenständiges wissenschaftliches Arbeiten auf dem Gebiet der Ingenieurinformatik oder Ingenieurwissenschaften | | |
| Arbeitsaufwand: Gesamt: 120 Std. 90 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium) 30 Std. Seminar (Präsenzstudium) | | |
| Voraussetzungen: Grundkenntnisse auf dem Gebiet der Regelungstechnik z.B. aus dem Bachelor-Studium. | | |
| Angebotshäufigkeit: jedes Sommersemester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 2 | Wiederholbarkeit: siehe PO des Studiengangs | |

| |
|---|
| Modulteile |
| Modulteil: Seminar zu speziellen Kapiteln der Regelungstechnik 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 Ingenieurinformatik 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 WS 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. |
| Zugeordnete Lehrveranstaltungen: Seminar zu speziellen Kapiteln der Regelungstechnik (Seminar) |

| |
|--|
| Prüfung Seminar zu speziellen Kapiteln der Regelungstechnik Seminar, Vortrag und schriftliche Ausarbeitung |
|--|

| | | |
|---|---|---|
| Modul INF-0003: Masterarbeit | | 30 ECTS/LP |
| Version 1.0.0 (seit WS13/14) Modulverantwortliche/r: Alle Professorinnen und Professoren, die Veranstaltungen für diesen Studiengang anbieten. | | |
| <p>Lernziele/Kompetenzen: Die Studierenden sind mit der wissenschaftlichen Methodik sowie Techniken der Literaturrecherche vertraut, sind in der Lage, unter Anleitung praktische oder theoretische Methoden zur Bearbeitung eines vorgegebenen Themas einzusetzen und besitzen die Kompetenz, ein Problem der Informatik innerhalb einer vorgegebenen Frist weitgehend selbständig mit wissenschaftlichen Methoden zu bearbeiten sowie die Ergebnisse schriftlich und mündlich darzustellen.</p> <p>Darüber hinaus können die Studierenden grundlegende Strategien, Methoden und klare Vorstellungen entwickeln über ihre Ziele und Prioritäten, sie bewerten ihren persönlichen Arbeitsstil und schaffen eine effiziente Nutzung ihrer Ressourcen. Sie wenden Hilfsmittel und Techniken der Selbstorganisation an, die ihrem persönlichen Arbeitsstil entsprechen.</p> <p>Schlüsselqualifikationen: Team- und Kommunikationsfähigkeit, Durchhaltevermögen, schriftliche und mündliche Darstellung eigener (praktischer oder theoretischer) Ergebnisse, Fertigkeit der effizienten Ressourcennutzung, Einschätzung der Relevanz eigener Ergebnisse, Grundsätze guter wissenschaftlicher Praxis, Fähigkeit zur Selbstreflexion und eines reflektierten Arbeitsstils, Fertigkeit der verständlichen, sicheren und überzeugenden (schriftlichen und mündlichen) Darstellung von eigenen und fremden (praktischen oder theoretischen) Ideen, Konzepten und Ergebnissen und deren Dokumentation</p> | | |
| <p>Arbeitsaufwand: Gesamt: 900 Std. 15 Std. Seminar (Präsenzstudium) 885 Std. Anfertigen von schriftlichen Arbeiten (Selbststudium)</p> | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: nach Bedarf | Empfohlenes Fachsemester: 4. | Minimale Dauer des Moduls: 1 Semester |
| SWS: 1 | Wiederholbarkeit: siehe PO des Studiengangs | |
| Modulteile | | |
| Modulteil: Masterarbeit | | |
| Sprache: Deutsch | | |
| SWS: 1 | | |
| 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 | | 0 ECTS/LP |
| Version 1.0.0 Modulverantwortliche/r: | | |
| Lernziele/Kompetenzen: Die Teilnehmer wissen, wie sie an wissenschaftliche Arbeiten heran gehen, welche Vorgehensweise sie ans Ziel führt und welche Maßstäbe gelten, damit ihre Arbeit als wissenschaftlich angesehen wird. | | |
| Bemerkung: Dies ist eine freiwillige Veranstaltung und gibt keine ECTS-Punkte! | | |
| Arbeitsaufwand: Gesamt: 15 Std. | | |
| Voraussetzungen: keine | | |
| Angebotshäufigkeit: jedes Semester | Empfohlenes Fachsemester: | Minimale Dauer des Moduls: 1 Semester |
| SWS: 1 | Wiederholbarkeit: keine | |
| Modulteile | | |
| Modulteil: Anleitung zu wissenschaftlichen Arbeiten Sprache: Deutsch SWS: 1 | | |
| Inhalte: Begleitung bei der Anfertigung von Seminar-/Bachelor-/Master-/Diplomarbeiten und Dissertationen. | | |

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