

Modulhandbuch

für die
Prüfungsordnungsversion
2021

Inhaltsverzeichnis

Agile Produktentwicklung.....	
Algorithmen und Datenstrukturen.....	
Computersicherheit.....	
Data Science.....	
Ingenieurwissenschaftliche Applikationen.....	
Künstliche Intelligenz und maschinelles Lernen.....	
Masterarbeit mit Kolloquium.....	
Netzwerkprogrammierung.....	
Numerische Mathematik.....	
Objektorientierte Programmierung - Vertiefung.....	
Parallele Programmierung.....	
Real Time Computing.....	
Rechner- und Netzwerkarchitektur.....	
Sicherheitskritische Systeme.....	
Software Engineering.....	
Systemprogrammierung.....	
Systems Engineering.....	
Wissenschaftliches Arbeiten.....	

Modulname	Modulcode
Agile Produktentwicklung	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Kai Mecke	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
keine
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
Agile, teamorientierte Arbeitsweise nach Scrum sowie div. Strategien für die Produktentwicklung - Systematische Produktentwicklung nach Pahl und Beitz, VDI 2221/2222 - Münchner Vorgehensmodell - Integrated Design Engineering sowie div. Strategien für die Softwareentwicklung (DevOps) - Software Lebenszyklus Betrachtungen - Planung & Entwicklung - Test-Driven Software development
Qualifikationsziele
In der heutigen dynamischen Unternehmensumwelt ist Agilität (Reaktionskompetenz) für Unternehmen überlebensnotwendig. Um ihre Reaktionsgeschwindigkeit zu verbessern, führen immer mehr Unternehmen agile Arbeitsweisen ein. Agile Arbeitsweisen bieten die Rahmenbedingungen für die Bewältigung komplexer Problemstellungen in einem interdisziplinären Team. Dabei steht ein effizientes und effektives Arbeiten im Team durch Selbst-Organisation und Transparenz zur Förderung von kreativen Lösungen im Vordergrund. Nach der erfolgreichen Teilnahme an dem Modul sind die Studierenden in der Lage - Gestellte Anforderungen von komplexen Aufgabenstellungen im Team zu analysieren und zu strukturieren.

- Ihre eigenen fachlichen Kompetenzen zu bewerten, sowie Lücken in Kompetenzen des Teams zu bewerten und ggf. zu kompensieren.
- Eine passende strategische Vorgehensweise für eine Problemlösung zu identifizieren
- Auf der Basis dieser Vorgehensweise eine teamorientierte Arbeitsplanung selbstverantwortlich zu generieren.
- Eigenständige Produkte bzw. Lösungen zu entwickeln, welche auf einer methodischen Produktentwicklung als auch auf einer methodischen Teamarbeit fundieren.

Lehr- und Lernmethoden

Vorlesung / Übung und Labor

Studentische Arbeitsbelastung

54h Kontaktzeit + 96h Selbststudium

Literatur

Rumpe, B.: Agile Modellierung mit UML. Springer, Berlin Heidelberg (2012)
Schatten, A., Demolsky, M., Winkler, D., Biffel, S., Gostischa-Franta, E., Östreicher, T. : Best Practice Software-Engineering. Springer, Berlin (2010)
Liebig H., Flik T., Rechenberg P., Reinefeld A., Mössenböck H.: Das Ingenieurwissen: Technische Informatik Springer Berlin (2014)
Schwaber K., Sutherland, J.: Der Scrum Guide www.scrum.org (2017)
Röpstorff, S., Wiechmann, R.: Scrum in der Praxis : Erfahrungen, Problemfelder und Erfolgsfaktoren. Dpunkt (2016)
Geisreiter M., Zuccaro C., Rambo J.: GfSE SE-Handbuch: Die Klammer in der technischen Entwicklung. Gesellschaft für Systems Engineering (2019)
Douglass, B. P.: Agile Systems Engineering. Morgan-Kaufmann (2016)

Modulname	Modulcode
Algorithmen und Datenstrukturen	
Modulverantwortliche/r	Einrichtung
Dr. Lars Nolle	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
2. Semester	in jedem Semester	1	PF	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Kenntnisse der Objektorientierten Programmierung
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
Statische und dynamische Datenstrukturen; elementare Datenstrukturen, zum Beispiel Listen, Stapel, Warteschlangen, Bäume, Graphen, Mengen; elementare Algorithmen, unter Anderem Sortieren, Traversieren, Suchen, Einfügen, Löschen, Optimieren, Hash-Funktionen; rekursive Algorithmen; deterministische und stochastische Algorithmen; parallele Algorithmen; Analyse von Algorithmen: Komplexität und asymptotisches Verhalten; praktische Übungen.
Qualifikationsziele
Nach erfolgreicher Teilnahme an dem Modul sind die Studierenden in der Lage, die wichtigsten Datenstrukturen und elementare Algorithmen zu erklären, in einer objektorientierten Hochsprache zu implementieren und einzusetzen. Sie können neue Algorithmen für die Lösung neuer Probleme entwickeln und untersuchen. Weiterhin sind die Studierenden in der Lage, Algorithmen hinsichtlich ihrer Komplexität und ihres asymptotischen Verhaltens zu analysieren und zu bewerten. Ferner können die Studierenden Algorithmen und selbst erstellten Quellcode testen und dokumentieren.
Lehr- und Lernmethoden
Vorlesung / Übung und Labor
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium

Literatur

Harel, D., Feldman, Y. (2004) Algorithmics: The Spirit of Computing, 3rd Ed., Addison-Wesley, ISBN 978-0321117847.

Malik, D. S. (2009) Data Structures using C++, 2nd Ed., Cengage Learning, ISBN 978-0324782011.

Deitel, P., Deitel, H. (2016) C++ How to Program, 10th Ed., Pearson, ISBN 978-0134448237.

Modulname	Modulcode
Computersicherheit	
Modulverantwortliche/r	Einrichtung
Dr. Lars Nolle	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Grundlagen der Informatik; Kenntnisse der objektorientierten Programmierung
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit
Lehrinhalte
Grundlegende Begriffe der Computersicherheit; Sicherheitsziele; Schwachstellen in Computersystemen und Netzwerken; Bedrohungen, Angriffe, Risiken und Risikoanalyse; Angriffsszenarien und Angreifer; Sicherheitsmechanismen und Gegenmaßnahmen, beispielsweise Checklisten, Kryptografie, Zugangskontrollen oder Intrusion Detection; Computerforensik und rechtliche Aspekte der Computersicherheit.
Qualifikationsziele
Nach erfolgreicher Teilnahme an der Veranstaltung sind die Studierenden in der Lage, Konzepte und Methoden der Computersicherheit auf Informationssysteme anzuwenden. Aufgrund rechtlicher Vorgaben können sie Sicherheitsziele, wie Vertraulichkeit, Integrität und Verfügbarkeit definieren und Schwachstellen in Computersystemen und Netzwerken identifizieren. Die Studierenden sind in der Lage, Risikoanalysen durchzuführen und geeignete Sicherheitsmechanismen, beispielsweise Verschlüsselung oder Zugangskontrollen gegen potentielle Angriffstechniken auszuwählen und deren Effektivität zu bewerten.
Lehr- und Lernmethoden
Vorlesung / Übung
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium
Literatur
Witt, B.C. (2006) IT-Sicherheit kompakt und verständlich : Eine praxisorientierte Einführung, Vieweg & Sohn, ISBN 9783834890771.

Spitz, S., Pramateftakis, M., Swoboda, J. (2011) Kryptographie und IT-Sicherheit : Grundlagen und Anwendungen, 2. u#berarbeitete Auflage, Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, ISBN 9783834881205.

Modulname	Modulcode
Data Science	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Alexandra Burger	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
2. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Programmierkenntnissen in einer höheren Sprache, empfohlen erfolgreiche Teilnahme am Modul „Künstliche Intelligenz und Maschinelles Lernen“
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
Dieses Modul konzentriert sich auf Konzepte, Methoden, Algorithmen und Werkzeuge für das Design, die Verwaltung und den Einsatz des „Knowledge Discovery from Data“ Prozesses. Das umfasst Predictive und Descriptive Data Mining Techniken. Insbesondere werden Sprachen und Tools zur Datenanalyse und zum Workflow-Management für praktische Aktivitäten an mehreren Testfällen eingesetzt. Die Teilnehmer lernen allgemeine Data Mining Prinzipien und Techniken kennen und wenden diese in verschiedenen Anwendungsdomänen an.
Qualifikationsziele
Nach erfolgreicher Teilnahme am Modul verstehen die Studierenden den allgemeinen Knowledge-Discovery-Prozess sowie die verschiedenen Data-Mining-Algorithmen und -Techniken. Sie sind in der Lage, diese in unterschiedlichen Kontexten anzuwenden und die Analyseergebnisse kritisch zu bewerten. Durch die praktischen Aktivitäten erlernten die Studierenden modernste Sprachen und Tools für die Implementierung von Data Science und Mining-Lösungen für verschiedenste Anwendungsdomänen und können sie anwenden.
Lehr- und Lernmethoden
Vorlesung / Übung und Labor
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium

Literatur

- Bramer, M. (2020) Principles of Data Mining, 4th Ed., Springer-Verlag London Ltd., ISBN 978-1-4471-7493-6.
- Data Mining, Concepts and Techniques, Second Edition Jiawei Han, Micheline Kamber Morgan Kaufmann Publishers, March 2006 ISBN 978-1-55860-901-3 ISBN 10:1-55860-901-6
- Dörn, Sebastian. " „Python lernen in abgeschlossenen Lerneinheiten “ ." (2020).

Modulname	Modulcode
Ingenieurwissenschaftliche Applikationen	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Alexandra Burger	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	in jedem Semester	1	PF	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
keine
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit
Lehrinhalte
<p>Die fachlichen Inhalte der Veranstaltung werden am Anfang des Semesters aus aktuellen Forschungsthemen des Fachbereiches und aus unterschiedlichen Disziplinen bestimmt und dienen der Inspiration der Studierenden, welche daraufhin eigene Inhalte definieren. Ein Fokus ist dabei der methodische Einsatz einer wissenschaftlichen Applikation (Analyse, Auswahl, Einsatz, Entwicklung...).</p> <p>Das Format der Veranstaltung ist bewusst dem verantwortlichen Dozenten überlassen. Denkbar ist eine Kombination aus Ringvorlesung mit aktuellen Fragestellungen von Industriepartnern und aktuellen Forschungsthemen (Vorstellung von aktuellen Veröffentlichungen) sowie Projektarbeit der Studierenden.</p>
Qualifikationsziele
<p>Die Studierenden setzen sich unter Berücksichtigung des Lehrformats „Forschendes Lernen“ mit einer ingenieurnahen Applikation der Informatik auseinander.</p> <p>Nach erfolgreicher Teilnahme am Modul haben die Studierenden eine forschende und anwendungsbezogene Haltung zu Applikationen der Informatik aus dem Ingenieurbereich entwickelt. Sie sind in der Lage, in diesem Themenfeld selbstständig Aufgabenstellungen zu identifizieren sowie Probleme und Herausforderungen systematisch zu bearbeiten. Weiter können die Studierenden eine ausgewählte Anwendung in vertiefter Weise eigenständig erschließen, darstellen und bewerten sowie mögliche Herausforderungen, Trends und Entwicklungen ableiten.</p>

Die Motivation der Studierenden wird durch die Vorstellung von aktuellen Forschungstätigkeiten am Fachbereich angeregt.

Ein mögliches Ergebnis der Veranstaltung für den Studierenden ist ein eigenes Forschungsthema, welches bis zu einer Antragsreife gebracht werden kann und ggf. einen Ausgangspunkt für eine fokussierte Entwicklung in einem Forscherteam (z.B. im Rahmen der agilen Produktentwicklung) darstellt. Weiterhin ist eine Vertiefung im Form von Masterarbeiten ein weiterer Anknüpfungspunkt.

Lehr- und Lernmethoden

Vorlesung / Übung

Studentische Arbeitsbelastung

54h Kontaktzeit + 96h Selbststudium

Literatur

Fachlich: Wissenschaftliche, aktuelle Literatur welche thematisch am Anfang des Semesters identifiziert wird.

Methodisch: Wulf, C., Haberstroh, S., Petersen, M., 2020. Forschendes Lernen: Theorie, Empirie, Praxis. Springer Fachmedien Wiesbaden.

Modulname	Modulcode
Künstliche Intelligenz und maschinelles Lernen	
Modulverantwortliche/r	Einrichtung
Dr. Lars Nolle	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Kenntnisse der Objektorientierten Programmierung, empfehlende Grundkenntnisse der numerischen Mathematik und Statistik
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
Grundbegriffe der Künstlichen Intelligenz; geschichtliche Entwicklung der Künstlichen Intelligenz; Superintelligenz und ethische Probleme; starke und schwache Künstliche Intelligenz, symbolische und nicht-symbolische Wissensrepräsentation; Einsatzgebiete der Künstlichen Intelligenz: Planen, Schließen, Suchen, Optimieren, Approximieren; ausgewählte Methoden der Künstlichen Intelligenz; Grundlagen des Maschinellen Lernens: überwachtes und unüberwachtes Lernen, verstärkendes und tiefes Lernen; Datenaufbereitung sowie wissenschaftliche Auswertung und Darstellung von Ergebnissen; praktische Übungen.
Qualifikationsziele
Nach erfolgreicher Teilnahme an dem Modul sind die Studierenden in der Lage, Künstliche Intelligenz und ihre Teilgebiete, speziell das Maschinelle Lernen, zu definieren und zwischen starker und schwacher Künstlicher Intelligenz zu unterscheiden. Studierende können Chancen und Risiken der Künstlichen Intelligenz und des Maschinellen Lernens benennen und mögliche Anwendungsfälle identifizieren. Die Studierenden können nach erfolgreichem Abschluss des Moduls für praktische Problemstellungen geeignete Methoden auswählen, implementieren, anwenden und kritisch bewerten. Weiterhin sind die Studierenden in der Lage, Ergebnisse von Untersuchungen wissenschaftlich auszuwerten und zu präsentieren.
Lehr- und Lernmethoden
Vorlesung / Übung und Labor
Studentische Arbeitsbelastung

54h Kontaktzeit + 96h Selbststudium

Literatur

Ertel, W. (2016) Grundkurs Künstliche Intelligenz - Eine praxisorientierte Einführung, 4. überarbeitete Auflage, Springer Vieweg, ISBN 978-3-658-13548-5.

Bramer, M. (2020) Principles of Data Mining, 4th Ed., Springer-Verlag London Ltd., ISBN 978-1-4471-7493-6.
--

Sharp, J.A., Peters, J., Howard, K. (2002) The Management of a Student Research Project, 3rd Ed., Gower Publishing, ISBN 978-0566084904.
--

Modulname	Modulcode
Masterarbeit mit Kolloquium	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Alexandra Burger	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
3. Semester	in jedem Semester	1	PF	30	

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Zur Masterarbeit wird zugelassen, wer mindestens 50 Leistungspunkte aus den Theoriesemestern nachweist.
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Masterarbeit und Kolloquium
Lehrinhalte
Die Masterarbeit ist eine selbständige wissenschaftliche Arbeit unter Anleitung zum Abschluss des Studiums zu einem komplexen und umfangreichen technischen Thema, die sorgfältig geplant, erfolgreich durchgeführt und angemessen dokumentiert werden muss.
Qualifikationsziele
Das im Studiengang erworbene vertiefte theoretische Wissen soll an einer anwendungsorientierten Aufgabenstellung aus einem aktuellen Forschungs- und Entwicklungsprojekt angewendet werden.
Studentische Arbeitsbelastung
Oh Kontaktzeit + 900h Selbststudium

Modulname	Modulcode
Netzwerkprogrammierung	
Modulverantwortliche/r	Einrichtung
Dipl.-Ing. Olaf Fischer Dipl.-Ing. Udo Willers	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Kenntnisse Grundlagen der Informatik, Hochsprachenprogrammierung
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
Grundlagen und Grundbegriffe der Netzwerk- und Übertragungstechnik; Netzwerktopologien und -klassen Normen und Spezifikationen; Schichtenmodelle und Protokolle Typische Applikationsprotokolle und deren Einsatzgebiet und Funktionsweise; Netzwerkanalyse Programmierung von Netzwerkanwendungen auf Basis der Socket- und Websocket-API praktische Übungen.
Qualifikationsziele
Nach erfolgreicher Teilnahme an dem Modul kennen die Studierenden den Aufbau und die Funktion der wichtigsten Kommunikationsmechanismen des ISO/OSI Referenzmodell und ihre Implementierung in beispielhaften Protokollen. Sie sind in der Lage komplexe Kommunikationsabläufe auf Protokollebene zu analysieren und Fehlerbilder mit Hilfe von Softwaretools zu bewerten. Studierende sind in der Lage Netzwerkanwendungen auf Basis von Standardschnittstellen zu entwickeln und zu testen. Sie können Netzwerkanwendungen eigenständig konzeptionieren und die Eignung von Protokollen und Programmierschnittstellen für bestimmte Anwendungsszenarien bewerten.
Lehr- und Lernmethoden
Vorlesung / Übung und Labor
Studentische Arbeitsbelastung

54h Kontaktzeit + 96h Selbststudium
Literatur
D. E. Comer, „Internetworking with TCP/IP“ Vol. I, Prentice-Hall D.E.Comer, D.L. Stevens Internetworking with TCP/IP, Vol. III: Client-Server Programming and Applications, Linux/Posix Sockets Version, Prentice-Hall M. Zahn "Unix-Netzwerkprogrammierung mit Threads, Sockets und SSL", Springer

Modulname	Modulcode
Numerische Mathematik	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Ammar Memari	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	in jedem Semester	1	PF	5	4

Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit
Lehrinhalte
In dieser Vorlesung werden grundlegende Bausteine behandelt, die für viele ingenieurwissenschaftliche Problemstellungen relevant sind. Dazu gehören Zahlendarstellung im Rechner, Kondition und Stabilität von Algorithmen, lineare Gleichungssysteme, LR- und QR-Zerlegung von Matrizen, Interpolation, Methode der kleinsten Fehlerquadrate, Numerische Integration und Integration gewöhnlicher Differentialgleichungen.
Qualifikationsziele
Die Studierenden verstehen nach Bestehen des Moduls Grundlagen der numerischen Lösung von Problemstellungen mit dem Computer. Sie können die zahlenmäßigen Ergebnisse von numerischen Algorithmen qualitativ bewerten. Grundlegende numerische Verfahren können sie sowohl händisch als auch mit einem Computer durchführen und auf naturwissenschaftliche-technische Problemstellungen anwenden. Am Ende dieses Kurses verstehen die Studierenden die Theorie von Algorithmen wie LR und QR-Zerlegung und sind in der Lage, sie beispielsweise für die Berechnung der kleinsten Fehlerquadrate anzuwenden. Sie verstehen mehrstufige Runge-Kutta-Verfahren zur Lösung gewöhnlicher Differentialgleichungen.
Lehr- und Lernmethoden
Vorlesung / Übungen
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium
Literatur
[1] W. Dahmen, A. Reusken: Numerik für Ingenieure und Naturwissenschaftler, Springer-Verlag, 2008, ISBN-13 978-3-540-25544-4 [2] P.Deuffhard, A.Hohmann: Numerische Mathematik I: Eine algorithmisch orientierte Einführung, de Gruyter-Verlag Berlin u.a. 2002, ISBN 3-11-017182-1 [3] R.Schaback, H.Wendland: Numerische Mathematik, Springer-Verlag Berlin u.a. 2005, ISBN 3-540-21394-5

[4] G. Stoyan, A. Baran: Elementary Numerical Mathematics for Programmers and Engineers, Springer-Verlag Berlin u.a. 2016, ISBN:3-319-44659-2

Modulname	Modulcode
Objektorientierte Programmierung - Vertiefung	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Ammar Memari	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Kenntnisse der objektorientierten Programmierung
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
Vertiefung von objektorientierten Konzepten und Methoden; Grundlagen der parallelen Programmierung; ereignisorientierte Programmierung und Grafische Benutzeroberflächen; Model-View-Controller Architektur; praktische Übungen mit einer objektorientierten GUI-Bibliothek, beispielsweise Qt; praktische Übungen und Anwendungen, zu Beispiel Parser und Reguläre Ausdrücke.
Qualifikationsziele
Nach erfolgreicher Teilnahme an der Veranstaltung sind die Studierenden in der Lage, weiterführende objektorientierte Konzepte und Methoden anzuwenden. Hierzu gehören beispielsweise die parallele und die ereignisorientierte objektorientierte Programmierung. Die Studierenden können Grafische Benutzeroberflächen für objektorientierte Programme entwickeln, testen und evaluieren. Sie sind in der Lage, ihre Programme und Ergebnisse professionell zu dokumentieren.
Lehr- und Lernmethoden
Vorlesung / Übung und Labor
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium

Literatur

Breyman, U. (2011) Der C++-Programmierer: C++ lernen – Professionell anwenden – Lösungen nutzen, 2. aktualisierte Auflage, Carl Hanser Verlag GmbH & Co. KG, ISBN 978-3-446-42691-7.

Blanchette, J., Summerfield, M. (2006) C++ GUI Programming with Qt 4, Prentice Hall International, ISBN 0131872494.

Modulname	Modulcode
Parallele Programmierung	
Modulverantwortliche/r	Einrichtung
Dr. Lars Nolle	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Kenntnisse der objektorientierten Programmierung
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
Grundbegriffe der parallelen Datenverarbeitung; Architekturen für Parallelrechner; Klassifizierung nach Flynn: SISD, MISD, SIMD, MIMD; Parallelisierbarkeit von Algorithmen; parallele Programmiermodelle, Message-Passing-Programmierung, Thread Programmierung, GPU Programmierung; Laufzeitanalyse paralleler Programme; Entwicklungsumgebungen für paralleles Programmieren, beispielsweise MPI, Pthreads, OpenMP, OpenCL oder Cuda; ausgewählte Anwendungsbeispiele aus dem ingenieurwissenschaftlichen Bereich.
Qualifikationsziele
Nach erfolgreicher Teilnahme an der Veranstaltung sind die Studierenden in der Lage, die Parallelisierbarkeit von Algorithmen zu bewerten. Sie können für ingenieurwissenschaftliche Problemstellungen parallele Algorithmen entwickeln, diese auf einer gegebenen Parallelrechnerarchitektur unter Zuhilfenahme geeigneter Programmiermodelle und Entwicklungsumgebungen implementieren und deren Laufzeitverhalten analysieren.
Lehr- und Lernmethoden
Vorlesung / Übung und Labor
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium
Literatur
Rauber, T., Gudula, R., Gudula, R. (2012) Parallele Programmierung, Springer, ISBN 9783642136030.

Modulname	Modulcode
Real Time Computing	
Modulverantwortliche/r	Einrichtung
Dr. rer. nat. Juliane Benra	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	in jedem Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
keine
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
<p>Definition von Echtzeitproblemen Besonderheiten von Echtzeitbetriebssystemen Synchronisation und Konsistenz von Echtzeitsystemen:</p> <ul style="list-style-type: none"> - Gegenseitiger Ausschluss - Kooperation - Leser-Schreiber-Problem - Produzenten-Konsumenten-Problem - Problem der speisenden Philosophen - Verwendung von Semaphoren <p>Programmierung von Echtzeitsystemen und die Notwendigkeit zur Verwendung spezieller Sprachen dafür Analyse und Design von Echtzeitsystemen Sicherheits-, Qualitäts- und Leistungsmessung in Echtzeitsysteme</p>
Qualifikationsziele
<p>Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage, Echtzeitsoftware zu entwickeln. Sie erkennen die Problematik, die in der Abarbeitung paralleler Tasks liegt und können die verschiedenen Problemstellungen identifizieren und mit geeigneten Methoden/Werkzeugen entgegensteuern. Sie wissen um die Vorzüge der Nutzung besonderer Umgebungen im Bereich der Echtzeitsysteme hinsichtlich der Nutzung von Betriebssystemen und Programmiersprachen.</p>

Lehr- und Lernmethoden
Vorlesung / Übungen und Labor
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium
Literatur
Benra/Halang (Hrsg.) Software-Entwicklung für Echtzeitsysteme ; Springer 2009 ergänzend: Brinkschulte/Wörn: Echtzeitsysteme; Springer 2005 Friedrich/Kienzle: Programmierung von Echtzeitsystemen; Hanser 2008 Zöbel: Echtzeitsysteme: Grundlagen der Planung; Springer 2008

Modulname	Modulcode
Rechner- und Netzwerkarchitektur	
Modulverantwortliche/r	Einrichtung
Dipl.-Ing. Olaf Fischer Dipl.-Ing. Udo Willers	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Kenntnisse Grundlagen der Informatik, Hochsprachenprogrammierung
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit
Lehrinhalte
Prozessorarchitekturen; Klassen von Rechnersystemen (Hostsysteme, Eingebettete Systeme, SoC, Mehrprozessorsysteme); Technologien zur Rechnervernetzung; Kommunikationsprotokolle; Benchmarking und Energieeffizienz; Beispielhafte Synthese eines vernetzten Rechnersystems zur Lösung einer Ingenieurwissenschaftlichen Problemstellung; praktische Übungen.
Qualifikationsziele
Nach erfolgreicher Teilnahme an dem Modul kennen die Studierenden Aufbau und Besonderheiten aktueller Rechnerarchitekturen sowie ihre typischen Einsatzszenarien in unterschiedlichen, ingenieurwissenschaftlichen Anwendungsdomänen. Sie verfügen weiterhin über Kenntnisse im Aufbau von Rechnernetzen unter Verwendung unterschiedlicher Technologien und Topologien. Studierende sind in der Lage, Systeme hinsichtlich ihrer Rechenleistung, ihres Datendurchsatz und Energiebedarfs zu analysieren, kritisch zu bewerten und auszuwählen. Am Ende des Moduls sollen die Studierenden in der Lage sein, ein komplexes System von vernetzten Rechnern zur Lösung einer ingenieurwissenschaftlichen Aufgabenstellung eigenständig zu synthetisieren.
Lehr- und Lernmethoden
Vorlesung / Übung
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium

Literatur

Andrew S. Tanenbaum und Todd Austin, Rechnerarchitektur (2014) 6. Auflage, Pearson, ISBN 978-3868942385

Andriy Luntovskyy, Dietbert Gütter, Moderne Rechnernetze (2020), Springer Vieweg, ISBN 978-3658256166

Modulname	Modulcode
Sicherheitskritische Systeme	
Modulverantwortliche/r	Einrichtung
Dr. Lars Nolle	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Kenntnisse der objektorientierten Programmierung, Grundkenntnisse der Softwaretechnik
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit
Lehrinhalte
Sicherheitsanforderungen an Cyber-physikalische Systeme; Fallstudien; Sicherheitsanalysetechniken wie zum Beispiel PAAG, FMECA oder Fault Tree Analysis; Architekturentwurf von sicherheitskritischen Systemen; Entwicklung sicherheitskritischer Systeme; formaler Tests; Software-Inspektion und Bedrohungsanalyse; organisatorische, kognitive und soziale Aspekte von Entwicklungsprojekten für sicherheitskritische Systeme; Sicherheitsnormen, beispielsweise IEC 61508; praktische Übungen mit einer Programmiersprache für sicherheitskritische Systeme, beispielsweise ADA oder SPARK.
Qualifikationsziele
Nach erfolgreicher Teilnahme an der Veranstaltung sind die Studierenden in der Lage, Cyber-physikalische Systeme hinsichtlich gegebener Sicherheitsanforderungen zu analysieren. Hierzu verwenden sie standardisierte Analysetechniken. Sie sind in der Lage, sichere Systeme gemäß internationaler Standards und Normen zu entwickeln und zu verifizieren. Des Weiteren können die Studierenden entsprechende Dokumentationen erstellen.
Lehr- und Lernmethoden
Vorlesung / Übung
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium
Literatur
Wolf, M. C., Serpanos, D. (2019) Safe and Secure Cyber-Physical Systems and Internet-of-Things Systems, Springer, ISBN 978-3-030-25808-5.

McCormick, J.W., Chapin, P.C. (2015) Building High Integrity Applications with SPARK, Cambridge University Press, ISBN 1107040736.

Modulname	Modulcode
Software Engineering	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Kai Mecke	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Grundlagen der Informatik; Kenntnisse der objektorientierten Programmierung
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
Requirements Engineering; Softwremetriken und Aufwandsabschätzung; Vorgehensmodelle und Entwicklungsprozesse; Softwarearchitekturen und Entwurfsmuster; Validierung und Verifikation; Softwarequalität und Qualitätsmanagement; Konfigurationsmanagement; Testgetriebene Softwareentwicklung und Kontinuierliche Integration.
Qualifikationsziele
Nach erfolgreicher Teilnahme an der Veranstaltung sind die Studierenden in der Lage, Kundenanforderungen an Software zu erheben, zu analysieren, zu spezifizieren und zu validieren. Sie können geeignete Vorgehensmodelle für ein Softwareentwicklungsprojekt auswählen und anwenden. Die Studierenden sind in der Lage, adäquate Softwarearchitekturen und Entwurfsmuster auszuwählen und zu implementieren. Hierbei verwenden sie moderne Entwicklungswerkzeuge wie beispielsweise Konfigurationsmanagement, Testgetriebene Softwareentwicklung und Kontinuierliche Integration. Die Studierenden sind in der Lage, die entwickelte Software zu validieren und zu dokumentieren.
Lehr- und Lernmethoden
Vorlesung / Übung und Labor
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium

Literatur

Balzert, H. (2011) Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, 3. Auflage, Spektrum Akademischer Verlag, ISBN: 3827417066.

Balzert, H. (2009) Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, 3. Auflage, Spektrum Akademischer Verlag, ISBN 9783827422477.

Modulname	Modulcode
Systemprogrammierung	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Ammar Memari	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
Die erfolgreiche Teilnahme an den Modulen Grundlagen der Informatik und Hochsprachenprogrammierung ist empfehlenswert.
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit Studienleistung: experimentelle Arbeit
Lehrinhalte
Darstellung der Struktur und Konzepte von Betriebssystemen, Bedienung von gängigen Hilfsprogrammen als Schnittstelle zum BS und Skriptprogrammierung in der Shell am Beispiel von bash. Programmierschnittstellen von BS Abstraktionen: Prozesskonzept und -management, Nachrichtwarteschlangen, Dateisysteme und -verwaltung und Pipes/Shared Memory. Besonderheiten von Multiprozessorsystemen, Virtualisierung und Containerisierung. Rust als moderne Systemprogrammiersprache.
Qualifikationsziele
Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage, eigene techniknahe Computerprogramme zu entwickeln. Sie verfügen über Kenntnisse der Struktur generischer Betriebssysteme am Beispiel von Linux. Sie kennen die Shell und können sie als Schnittstelle zum unterliegendem BS effizient benutzen. Die Studierenden kennen und verstehen Abstraktionen, die das BS anbietet: Prozesse, Nachrichtwarteschlangen, Dateisysteme und Pipes/Shared Memory und sind in der Lage, deren Programmierschnittstellen zu untersuchen. Dadurch sind die Studierenden in der Lage, eigene Systemprogramme in C zu entwickeln, die diese Schnittstellen benutzen. Die Studierenden können die Notwendigkeit und Besonderheiten von Multiprozessorsystemen beschreiben und analysieren und verfügen über die Befähigung zur Zusammenarbeit mit Softwareentwicklern. Die Studierenden verstehen Containers und können sie benutzen am Beispiel von Docker. Sie haben am Ende des Kurses die Programmiersprache Rust als mögliche moderne Alternative

zu C für Systemprogrammierung kennen gelernt und sind in der Lage, ihre Eigenschaften zu definieren.

Lehr- und Lernmethoden

Vorlesung / Übung und Labor

Studentische Arbeitsbelastung

54h Kontaktzeit + 96h Selbststudium

Literatur

Tane- Tannenbaum, Andrew S., und Herbert Bos. Moderne Betriebssysteme. Pearson Deutschland GmbH, 2016.

Erich, Ehses, Köhler Lutz, Riemer Petra, Stenzel Horst, und Victor Frank. Systemprogrammierung in UNIX / Linux: Grundlegende Betriebssystemkonzepte und praxisorientierte Anwendungen. Vieweg+Teubner Verlag, 2012.

MandMandl, Peter. Grundkurs Betriebssysteme: Architekturen, Betriebsmittelverwaltung, Synchronisation, Prozesskommunikation, Virtualisierung. 4. Aufl. Springer Vieweg, 2014.

Silbe- Silberschatz, Abraham, Greg Gagne, und Peter B. Galvin. Operating System Concepts, 10th Edition. Wiley, 2018.

Modulname	Modulcode
Systems Engineering	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Alexandra Burger	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	jedes 2.Semester	1	WP	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
keine
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit
Lehrinhalte
<p>Die Inhalte von „Systems Engineering“ sollen an das disziplinspezifische Vorwissen der Studierenden anknüpfen, daher Wiederholung disziplinspezifischer Grundlagen, zum Beispiel aus Maschinenbau, Elektrotechnik, Informatik und weitere, um diese dann im Sinne des System Engineerings (Tugenden, Tätigkeiten und Hilfsmittel) zusammenzuführen. Schwerpunkte sind:</p> <ul style="list-style-type: none"> • Systemtheorie; • Modellbildung (Systemanalyse, -funktionen, -architektur, -entwurf, Integration, Verifikation und Validierung); • SysML; • Model-Based Systems Engineering. <p>Begleitende Inhalte: Vorgehensmodelle für die verteilte Produktentwicklung, Produktlebenszyklusmodelle, Wirtschaftliche Betrachtung.</p>
Qualifikationsziele
<p>Systems Engineering ist ein methodischer, multidisziplinärer Ansatz für das Design, die Realisierung, das technische Management, den Betrieb und die Stilllegung eines Systems. Ein „System“ ist die Kombination von Elementen, die zusammenarbeiten, um die Fähigkeit zu erzeugen, die zur Erfüllung eines Bedarfs erforderlich ist (Systemtheoretische Sichtweise auf ein komplexes Produkt).</p> <p>Nach der erfolgreichen Teilnahme an dem Modul können die Studierenden</p>

- den Begriff und das Fachvokabular des Systems Engineering erklären
- die disziplinspezifischen Grundlagen anwenden
- systemtheoretische Modellbildung auf ausgewählte Problemstellungen durchführen
- komplexe, disziplinübergreifende Systeme kombinieren, modellieren und simulieren (SysML und MBSE)
- fremde Modelle konstruktiv beurteilen, auswählen und in eigene Lösungen integrieren.

Lehr- und Lernmethoden

Vorlesung / Übung

Studentische Arbeitsbelastung

54h Kontaktzeit + 96h Selbststudium

Literatur

- Schatten, A., Biffel, S., Demolsky, M., Gostischa-Franta, E., Östreicher, T., Winkler, D., 2010. Best Practice Software-Engineering: Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen. Springer-Verlag.
- Weilkiens, T., 2014. Systems Engineering mit SysML/UML: Anforderungen, Analyse, Architektur. Mit einem Geleitwort von Richard Mark Soley, EBL-Schweitzer. dpunkt.verlag.
- Wulf, C., Haberstroh, S., Petersen, M., 2020. Forschendes Lernen: Theorie, Empirie, Praxis. Springer Fachmedien Wiesbaden.
- Zuccaro, C., Rambo, J., Hüffer, H., Fritz, J., Dorsch, T., 2019. GfSE SE-Handbuch: Die Klammer in der technischen Entwicklung. BoD – Books on Demand.

Modulname	Modulcode
Wissenschaftliches Arbeiten	
Modulverantwortliche/r	Einrichtung
Dr.-Ing. Alexandra Burger	Fachbereich Ingenieurwissenschaften

Semester	Angebotsfrequenz	Dauer	Modulart	Leistungspunkte	SWS
1. Semester	in jedem Semester	1	PF	5	4

Studiengänge
Ingenieurinformatik, M. Sc.

Voraussetzungen (für die Teilnahme)
keine
Lehrsprache
deutsch
Prüfungsform / Prüfungsdauer
Prüfungsleistung: Klausur 1,5h o. mündliche Prüfung o. Kursarbeit
Lehrinhalte
Einführung in das wissenschaftliche Arbeiten; Qualitätsanforderungen und ethische Aspekte von Forschungsvorhaben; Planung und Durchführung von Forschungsprojekten; Formulierung von Forschungsfragen und Einordnung in den Stand der Wissenschaft; Literaturrecherche in Datenbanken; Zitiertechniken; wissenschaftliche Methodik und Argumentation; Dokumentation, Aufbereitung und Darstellung von Daten und Ergebnissen; Erstellung wissenschaftlicher Texte; Konzeption und Durchführung von Vorträgen.
Qualifikationsziele
Nach erfolgreicher Teilnahme an der Veranstaltung sind die Studierenden in der Lage, Forschungsvorhaben zu planen und durchzuführen. Sie können eigene Forschungsfragen formulieren und sich eigenständig den Stand der Wissenschaft erarbeiten, indem sie Literaturquellen in gängigen wissenschaftliche Datenbanken suchen, die sie anschließend bewerten und zitieren. Sie können wissenschaftliche Methoden zur Beantwortung der Forschungsfragestellungen entwickeln, bewerten und anwenden. Sie sind in der Lage, qualifizierte Aussagen über die Qualität die Ergebnisse der Forschungsarbeit zu treffen und diese Ergebnisse adäquat zu kommunizieren. Hierzu erstellen sie wissenschaftliche Texte und Vorträge gemäß formaler Anforderungen.
Lehr- und Lernmethoden
Vorlesung / Übung
Studentische Arbeitsbelastung
54h Kontaktzeit + 96h Selbststudium

Literatur

Heesen, B. (2014) Wissenschaftliches Arbeiten - Methodenwissen für das Bachelor-, Master- und Promotionsstudium, Springer-Verlag Berlin Heidelberg, ISBN 978-3-662-43346-1.
