

Modulhandbuch Technische Informatik Bachelor of Science

Version B_TInf25.1_W

Letzte Änderung: 2026-07-11 15:25:33

Inhaltsverzeichnis

MB001 – Analysis
MB002 – Mathematische Konzepte und Diskrete Mathematik
MB362 – Study Bootcamp Technik
MB369 – Digitaltechnik
MB327 – Digitale Schaltungen und Prozessoren
MB344 – Objektorientierte Programmstrukturen
MB358 – Study Bootcamp Informatik
MB377 – Deskriptive Statistik und Lineare Algebra
MB032 – Übertragungstechnik
MB040 – Algorithmen und Datenstrukturen
MB043 – Systemnahe Programmierung
MB046 – Ingenieurmathematik
MB048 – Elektronik
MB378 – Fortgeschrittene Lineare Algebra
MB037 – Rechnernetze
MB057 – Fortgeschrittene Objektorientierte Programmierung
MB068 – Halbleiterschaltungstechnik
MB073 – Systemtheorie
MB101 – Echtzeitsysteme
MB368 – Venture Lab
MB108 – Großintegrierte Systeme
MB109 – Regelungstechnik
MB135 – Projekt Eingebettete Systeme
MB148 – Seminar Technische Informatik
MB366 – Datenbanken
MB373 – Robotik
MB118 – Soft Skills
MB122 – IT-Sicherheit
MB133 – Laborprojekt
MB134 – Diskrete Systeme
MB143 – Systementwurf mit VHDL
MB257 – Auslandssemester
MB150 – Bachelor-Thesis
MB370 – Bachelor-Kolloquium
MB371 – Praktikum

Module

◆ MB001 – Analysis

Verantwortliche:	Fikret Koyuncu
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB001 – Analysis
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	90 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 120 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Fikret Koyuncu

Teilleistung:	TB002 – Übg. Analysis
Lernform:	Übung
Prüfungsform:	Teilnahme
Prüfungsdauer:	
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 15 Stunden weicht signifikant von erwarteten 2.0 ECTS ab.</i>
Lehrende:	Fikret Koyuncu

Studieninhalte:

- Bearbeitung von Übungsaufgaben aus dem Themenspektrum der zugehörigen Lehrveranstaltung
- Vorstellung und Diskussion möglicher Lösungswege

-
- Zahlentypen
 - Folgen
 - Bildungsgesetze
 - Grenzwerte
 - Funktionen, Relationen
 - Funktionstypen
 - Umkehrfunktion
 - Differentialrechnung
 - Differentiationsregeln
 - Anwendungen der Differentialrechnung (Kurvendiskussionen und Extremwerte)
 - Unendliche Reihen
 - Integralrechnung
 - Integrationsmethoden
 - Anwendungen der Integralrechnung
 - Funktionen mit zwei Variablen
 - Partielle Differentiation
 - Extremwertaufgaben ohne Nebenbedingungen

Lernergebnisse:

Die Studierenden können ...

- praktische Problemstellungen mathematisch formulieren
- beurteilen, welche analytischen Hilfsmittel zielführend sind
- neue, unklare und ungewöhnliche Aufgabenstellungen als solche erkennen und mit weiterführender Hilfestellung bearbeiten
- Lösungsansätze präsentieren und begründen

Die Studierenden ...

- kennen und verstehen die grundlegenden Begriffe, Aussagen und Methoden der Analysis,
- können mathematische Regeln korrekt anwenden,
- verstehen Beweistechniken,
- erkennen die fundamentale Bedeutung des Grenzwertbegriffes für die Analysis,
- beherrschen die Methoden des Differenzierens und Integrierens,
- können die eindimensionale Differentialrechnung bei praxisorientierten Fragestellungen flexibel in unterschiedlichen Fachgebieten einsetzen und dabei beurteilen, welche analytischen Hilfsmittel für welche Problemstellungen zielführend sind
- erkennen die Anwendbarkeit und den Nutzen der Analysis für unterschiedliche Fachgebiete und deren spezifischen Problemstellungen,
- können praxisorientierte Problemstellungen in mathematische Beziehungen bzw. Modelle umzusetzen und anhand analytischer Modelle weiter bearbeiten
- können neue, unklare und ungewöhnliche Aufgabenstellungen als solche erkennen und zur Bearbeitung weiterführende Hilfestellung in Anspruch nehmen,
- verfügen über gesteigerte Kompetenzen sich Fähigkeit durch Selbststudium anzueignen und sich in neue formale Systeme einzuarbeiten

Voraussetzungen und Empfehlungen:

- Schulbildung in mathematischen Grundlagen
- Empfehlung: Brückenkurs Mathematik

Literatur:

- BÖHME, Gert:
Analysis 1.
6. Aufl. Berlin: Springer-Verlag, 1990
- FETZER, Albert; FRÄNKEL, Heiner:
Mathematik 1.
10. bearbeitete Aufl. Berlin: Springer-Verlag, 2008
- FETZER, Albert; FRÄNKEL, Heiner:
Mathematik 2.
6. korrigierte Aufl.. Berlin: Springer-Verlag, 2009
- HENZE, Norbert; Last, Günter:
Mathematik für Wirtschaftsingenieure 1.
2. Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2005
- KUSCH, Lothar:
Mathematik. Aufgabensammlung mit Lösungen. Bd. 3
9. Aufl. Berlin: Cornelsen Verlag, 1995
- OHSE, Dietrich: Mathematik für Wirtschaftswissenschaftler 1. Analysis.
6. Aufl. München: Verlag Vahlen, 2004
- PAPULA, Lothar:
Mathematik für Ingenieure und Naturwissenschaftler 1: Ein Lehr- und Arbeitsbuch für das

Grundstudium.

12. überarbeitete und erweiterte Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2009

- PREUSS, Wolfgang; WENISCH, Günter:
Lehr- und Übungsbuch Mathematik 1: Grundlagen - Funktionen - Trigonometrie.
2. neu bearbeitete Aufl. München: Carl Hanser Verlag, 2003
- PREUSS, Wolfgang; WENISCH, Günter:
Lehr- und Übungsbuch Mathematik 2: Analysis.
3. Aufl. München: Carl Hanser Verlag, 2003
- PAPULA, Lothar: Mathematik für Ingenieure und Naturwissenschaftler: Klausur- und Übungsaufgaben 4. überarbeitete und erweiterte Aufl. Wiesbaden: Vieweg + Teubner Verlag, 2010

Studiengänge:

- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB002 – Mathematische Konzepte und Diskrete Mathematik

Verantwortliche:	Sebastian Iwanowski
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB003 – Diskrete Mathematik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	5.0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden
Aufwand während Semesterferien:	30 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 120 Stunden weicht signifikant von erwarteten 5.0 ECTS ab.</i>
Lehrende:	Sebastian Iwanowski

Studieninhalte:

- Logik
 - Einführung
 - Aussagenlogik
 - Prädikatenlogik
- Mengenlehre
 - Grundlegende Begriffe und Konzepte
 - Relationen
 - Funktionen
 - Boolesche Algebren
- Beweisführung
 - Strukturen der mathematischen Beweisführung
 - Vollständige Induktion
 - Beweisstrategien
- Zahlentheorie
 - Teilbarkeit
 - Teilen mit Rest
 - Primzahlen
 - Modulare Arithmetik
- Algebraische Strukturen
 - Gruppen
 - Körper
- Kombinatorik
 - Zählformeln für Mengen
 - Permutationen
- Graphentheorie
 - Terminologie und Repräsentation
 - Wege in Graphen
 - Bäume
 - Planare Graphen
 - Färbungen

Lernergebnisse:

Nach Abschluss der Veranstaltung besitzen die Studierenden folgende Kompetenzen:

- Beherrschen der grundlegenden mathematischen Begriffe und Konzepte (Definition, Satz, Beweis) und Fähigkeit zur Unterscheidung derselben.
- Beherrschen der Grundlagen und der Formalisierung logischen Denkens.
- Verständnis elementarer Logik und Mengenlehre und des inneren Zusammenhangs dieser Gebiete.
- Darauf aufbauendes Verständnis von Relationen und Funktionen.
- Fähigkeit, elementare Beweisprinzipien wie vollständige Induktion in verschiedenen Kontexten anzuwenden.
- Beherrschen der grundlegenden Sätze der elementaren Zahlentheorie, Gruppen- und Körpertheorie, Kombinatorik und Graphentheorie und selbständige Anwendung an Beispielen.

Voraussetzungen und Empfehlungen:

Mathematik Gymnasium 9. Klasse

Literatur:

- Sebastian Iwanowski / Rainer Lang:
Diskrete Mathematik mit Grundlagen, Springer 2014, ISBN 978-3-658-07130-1 (Print), 978-3-658-07131-8 (Online)
- Albrecht Beutelspacher / Marc-Alexander Zschiegner:
Diskrete Mathematik für Einsteiger.
Vieweg 2004 (2. Auflage), ISBN 3-528-16989-3
- Norman L. Biggs:
Discrete Mathematics.
Oxford University Press 2002, ISBN 0-19-850717-8
- Neville Dean: Diskrete Mathematik.
Pearson Studium, Reihe "im Klartext" 2003, ISBN 3-8273-7069-8
- Christoph Meinel / Martin Mundhenk:
Mathematische Grundlagen der Informatik.
Teubner 2002 (2. Auflage), ISBN 3-519-12949-3

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB362 – Study Bootcamp Technik

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	2 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB356 – Study Bootcamp Technik - Projekt
Lernform:	Projektarbeit
Prüfungsform:	Portfolio-Prüfung
Prüfungsdauer:	75 Min.
ECTS:	8.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	34 Semesterwochenstunden
Termine im Semester:	7 Termine
Zeit in Veranstaltungen:	148 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 268 Stunden weicht signifikant von erwarteten 8.0 ECTS ab.</i>
Lehrende:	Jürgen Günther Ulrich Hoffmann

Teilleistung:	TB393 – Study Bootcamp Technik - Ingenieurskompetenz
Lernform:	Bootcamp
Prüfungsform:	Abnahme
Prüfungsdauer:	
ECTS:	7.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	18 Semesterwochenstunden
Termine im Semester:	7 Termine
Zeit in Veranstaltungen:	78 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	125 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	213 Stunden
Lehrende:	Dominik Miller Carsten Burmeister Andreas Haase Jürgen Günther

Studieninhalte:

Kreativitätstechniken und Brainstorming (Crazy 8s, Mind-Mapping); Grundlagen des Projektmanagements (Kanban, Scrum); Grundlagen der Programmierung für Mikrocontroller (MicroPython auf ESP32); Digitale Konstruktion (CAD) für Rapid Prototyping (FDM-3D-Druck, Laser-Cutter); Technisches Zeichnen; Grundlagen der Elektrotechnik; Grundlagen der Mechanik; Physikalisches Praktikum. Das Modul wird durch ein 6-wöchiges interdisziplinäres Teamprojekt (4 Personen) strukturiert, in dem ein smartes physisches Gerät konzipiert, entworfen und gefertigt wird. Wöchentliche Plenarversammlungen dienen der Fortschrittspräsentation; das Semesterergebnis wird abschließend öffentlich präsentiert.

Lernergebnisse:

Im Feld „Projektentwicklung“ können die Studierenden ...

- Ideen für ein stoffliches Produkt entwickeln und bewerten.
- eine Produktidee strukturiert konkretisieren und in eine umsetzbare Aufgabenstellung überführen.
- einen Prototypen durch technische Zeichnungen planen und praktisch fertigen.
- Projektarbeit koordinieren, Projektfortschritt dokumentieren und zielgruppengerecht präsentieren.

Im Feld „Technisches Zeichnen“ können die Studierenden ...

- technische Zeichnungen lesen und verstehen.
- einfache normgerechte Zeichnungen selbst erstellen.
- Toleranzen, Passungen und Oberflächengüten interpretieren.

- abhängig von der Zeichnung geeignete Bauteile auswählen.

Im Feld „CAD“ können die Studierenden ...

- grundlegende CAD-Funktionen sicher anwenden.
- sich selbständig in weitergehende CAD-Funktionen einarbeiten.
- Dateien für den 3D-Drucker und Laser-Cutter vorbereiten und exportieren.
- normgerechte CAD-Zeichnungen erstellen.

Im Feld „Programmierung“ können die Studierenden ...

- einfache Programme in MicroPython strukturieren und auf einem ESP32 ausführen.
- digitale und analoge Ein- und Ausgaben eines Mikrocontrollers ansteuern.
- Sensoren und Aktoren programmatisch einbinden und im Projektkontext nutzen.
- Fehler im Code systematisch identifizieren und beheben.

Im Feld „Elektrotechnik“ können die Studierenden ...

- grundlegende elektrische Größen (Spannung, Strom, Widerstand, Leistung) benennen und deren Zusammenhänge erläutern.
- einfache Gleichstromschaltkreise analysieren und dimensionieren.
- Schaltpläne lesen und einfache Schaltungen auf einem Steckbrett aufbauen und testen.
- typische elektronische Bauelemente (Widerstände, LEDs, Transistoren, Sensoren) identifizieren und fachgerecht einsetzen.

Im Feld „Mechanik“ können die Studierenden ...

- grundlegende mechanische Größen (Kraft, Drehmoment, Druck, Reibung) benennen und in einfachen Systemen anwenden.
- statische und kinematische Zusammenhänge an technischen Bauteilen und Konstruktionen qualitativ beurteilen.
- geeignete Materialien und Fertigungsverfahren für einfache mechanische Komponenten auswählen und begründen.
- die mechanische Machbarkeit eines Entwurfs anhand grundlegender Überlegungen einschätzen.

Im Feld „Physikalisches Praktikum“ können die Studierenden ...

- physikalische Messungen systematisch planen, durchführen und protokollieren.
- Messergebnisse auswerten, grafisch darstellen und interpretieren.
- Messunsicherheiten abschätzen und im Ergebnis berücksichtigen.
- den Bezug zwischen physikalischen Grundgesetzen und ihrer praktischen Relevanz im ingenieurtechnischen Kontext herstellen.

Voraussetzungen und Empfehlungen:

Keine formalen Voraussetzungen; das Modul ist explizit als Einstieg in das Studium konzipiert. Neugierde, Teamfähigkeit und die Bereitschaft zum praktischen Ausprobieren und zur öffentlichen Präsentation von Ergebnissen werden vorausgesetzt.

Literatur:

- Monk, S.: *Programming ESP32: Learn MicroPython Coding and Electronics*. MonkMakes, 2nd ed., 2024. ISBN 978-1-73948-745-4.
- Giesecke, F. E. et al.: *Technical Drawing with Engineering Graphics*. 15th ed. Pearson, Hoboken, 2016. ISBN 978-0-13-401152-5.

- Tipler, P. A., Llewellyn, R. A.: *Physik für Wissenschaftler und Ingenieure*. 7. Aufl. Springer Spektrum, Berlin, 2015. ISBN 978-3-642-54165-0.

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB369 – Digitaltechnik

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB069 – Prakt. Digitaltechnik
Lernform:	Praktikum
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	30 Min.
ECTS:	2,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	4 Termine
Zeit in Veranstaltungen:	10 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	4 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	46 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Thomas Starke

Teilleistung:	TB364 – Digitaltechnik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	3,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	5 Stunden
Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 75 Stunden weicht signifikant von erwarteten 3,0 ECTS ab.</i>
Lehrende:	Sergei Sawitzki

Studieninhalte:

- Einleitung: Digitale Systeme
- Mathematische Grundlagen
 - Entstehungsgeschichte
 - Aussagenlogik und Boolesche Algebra
 - Schaltalgebra, Schaltfunktionen und Schaltfunktionssysteme
 - Operatorensysteme
 - Normalformen und Dualitätsprinzip
- Schaltnetze
 - Darstellung
 - Vereinfachung (KV-Diagramme, QMCV, BDDs)
 - Analyse (Funktion, Komplexität, Zeitverhalten)
 - Synthese und Realisierung
 - Beispiele
- Speicherelemente
- Schaltnetzentwurf: Schaltnetz wird aus einer gegebenen Spezifikation formal entworfen. Der Entwurf wird auf einem IC-Trainer realisiert. Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. Die Ergebnisse werden dokumentiert.
- Schaltwerkentwurf: Schaltwerk (z. B. ein Zähler) wird aus einer gegebenen Spezifikation formal entworfen. Der Entwurf wird auf einem IC-Trainer realisiert. Die Schaltung wird auf Funktion und Einhaltung der Spezifikation überprüft. Die Ergebnisse werden dokumentiert.

Lernergebnisse:

Die Studierenden ...

- kennen die mathematischen Grundlagen des Schaltnetz-Entwurfs
- können darauf aufbauend einfache Schaltnetze entwerfen und optimieren
- erkennen Schaltnetze als technische Umsetzung von Schaltfunktionen
- kennen grundlegende Bausteine digitaler Systeme (Logikgatter, Multiplexer, Demultiplexer, arithmetische Schaltungen)
- können die Funktionsweise von einfachen Speicherelementen erläutern
- sind fähig, einfache digitale Systeme zu begreifen, zu spezifizieren, zu entwerfen und zu optimieren

Voraussetzungen und Empfehlungen:

Grundlegende Vorstellung von digitalen Rechnern

Literatur:

- Hoffmann, Dirk: Grundlagen der technischen Informatik, 5. Auflage, Carl Hanser Verlag 2016
- Schiffmann, Wolfram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Beuth, Klaus: Elektronik 4. Digitaltechnik, 13. Auflage Vogel Verlag und Druck 2003

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB327 – Digitale Schaltungen und Prozessoren

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB319 – Digitale Schaltungen, Prozessoren
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	150 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	5 Stunden
Flexibel einteilbarer Aufwand:	40 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 105 Stunden weicht signifikant von erwarteten 5,0 ECTS ab.</i>
Lehrende:	Sergei Sawitzki Dennis Säring

Studieninhalte:

- Rechnerarchitekturen (Entwicklung, Architekturkonzepte, parallele und nicht sequentielle Architekturen)
- Mikroprogrammierung
- Architekturkonzepte
- Mikroprogrammierung
- Mehrprozessorsysteme
- Aktuelle und zukünftige Entwicklungen
- Schaltwerke
 - Einleitung und Grundbegriffe, Definitionen
 - Speicherelemente
 - Analyse
 - Synthese
 - Zusammenschaltung
 - Transformationen
 - Zustandskodierung
 - Zustandsminimierung
 - Realisierung, Beispiele
- Zeitverhalten
 - Zeitverhalten von Schaltnetzen
 - Modellierung der Gatter- und Leitungsverzögerungen
 - Statische Timing-Analyse (STA)
 - Zeitverhalten von Schaltwerken
 - Metastabilität

Lernergebnisse:

Die Studierenden ...

- kennen grundlegende Rechnerarchitekturkonzepte, die beschreiben, wie verschiedene Baugruppen von Rechnern zusammenarbeiten und wie sich unterschiedliche Rechnersysteme voneinander unterscheiden
- kennen die Funktionselemente von Rechnern mit ihren typischen Systemeigenschaften und deren Abbildung auf ein bestimmtes Architekturmodell

- können das Zusammenwirken der beteiligten Hardware- und Softwarekonzepte im Rahmen einer Aufgabe zur Informationsverarbeitung einschätzen
- besitzen ein Verständnis für Ansätze zur Steigerung der Systemleistung insbesondere unter Berücksichtigung der Aspekte von Parallelität
- kennen Aufbau und Funktionsweise von Speicherelementen und Schaltwerken
- beherrschen die Methoden zur Analyse, Darstellung und Vereinfachung von endlichen Zustandsautomaten
- erkennen ein Schaltwerk als technische Umsetzung eines endlichen Zustandsautomaten
- beherrschen die Methoden der Zeitverhaltensanalyse und Zeitverhaltensoptimierung von digitalen Systemen, sowie können das Zeitverhalten und die Zeitvorgaben beim Entwurf digitaler Systemen berücksichtigen
- können digitale Systeme mittlerer Komplexität begreifen, spezifizieren, entwerfen und optimieren

Voraussetzungen und Empfehlungen:

Kenntnisse der wichtigsten Komponenten eines Rechnersystems.

Literatur:

- Martin: Einführung in die Rechnerarchitektur, Fachbuchverlag Leipzig, 2003
- Oberschelp, Gossen: Rechneraufbau und Rechnerstrukturen, Verlag Oldenbourg 1998
- van de Goor: Computer Architecture and Design, Verlag Addison Wesley, 1989
- Müller-Schloer, Schmitter: RISC-Workstation Architekturen, Verlag Springer 1991
- Ungerer: Datenfluß-Rechner, Verlag Teubner, 1993
- Hoffmann, Dirk: Grundlagen der technischen Informatik, Carl Hanser Verlag 2007
- Schiffmann, Wolram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Rabaey, Jan; Chandrakasan, Anantha; Nikolic, Borivoje: Digital Integrated Circuits, A Design Perspective, 2nd edition, Prentice Hall 2003
- Beuth, Klaus: Elektronik 4. Digitaltechnik, 13. Auflage, Vogel Verlag und Druck 2003

Studiengänge:

- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB344 – Objektorientierte Programmstrukturen

Verantwortliche:	Dennis Proppe
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB338 – Objektorientierte Programmstrukturen
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	120 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	15 Stunden
Aufwand während Semesterferien:	15 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Dennis Proppe

Teilleistung:	TB383 – Übg. Objektorientierte Programmstrukturen
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	30 Min.
ECTS:	3.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	22 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	50 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	82 Stunden
Lehrende:	Gerit Kaleck

Studieninhalte:

Es wird in die Programmierung mit Java und die Entwicklungsumgebung IntelliJ eingeführt. In der Übung werden die in der Vorlesung vorgestellten Grundkonzepte der objektorientierten Programmierung durch das Lösen verbal formulierter Aufgabenstellungen in kleinen Teams angewendet. Das Testen und Präsentieren sauber strukturierter Lösungen wird geübt.

Behandelte Grundkonzepte sind:

- Grundkonzept der Programmiersprache Java
 - Grundlegende Eigenschaften der Sprache
 - Grundlegender Aufbau von Java-Programmen
 - Ausführung von Java-Programmen
- Grundlegende Programmelemente
 - Primitive Datentypen in Java
 - Variablen, Zuweisung, Gültigkeitsbereiche
 - Operatoren und Ausdrücke
 - Anweisungen
- Referenzdatentypen
 - Arrays
 - Klassen
- Statische Methoden
- Grundlegende Klassen
 - String
 - StringBuilder
 - Wrapper-Klassen für primitive Datentypen
 - Enum
- Grundkonzepte der Objektorientierung
 - Klassen und Instanzen mit Attributen und Methoden
 - Sichtbarkeit, Packages
 - Konstruktoren
 - Vererbung und Überschreiben
 - Dynamisches Binden, Polymorphie
 - Objektorientierte Realisierung rekursiver dynamischer Datenstrukturen (Listen)
 - Generische Typen
 - Abstrakte Klassen und Interfaces - Deklaration und Nutzung
 - Realisierung grafischer Benutzungsoberflächen

- Behandlung von Laufzeitfehlern
- Klassen zur Realisierung von Dateioperationen

Lernergebnisse:

Die Studierenden ...

- identifizieren die Basiskonzepte der Objektorientierten Programmierung und stellen diese den Konzepten der prozeduralen Programmierung gegenüber.
- entwickeln Software auf der Grundlage der Kernkonzepte der Objektorientierten Programmierung.
- stellen die grundlegenden Sprachelemente (Datentypen, Anweisungen, Realisierung von objektorientierten Konzepten) von Java zusammen und wählen daraus aus, um Java-Programme mittlerer Komplexität zu entwickeln.
- vergleichen die Programmiersprachen Pascal und Java und stellen ihre Gemeinsamkeiten und Unterschiede heraus.
- setzen eine moderne Entwicklungsumgebung zur Unterstützung der Softwareentwicklung ein und stellen die damit verbundenen Funktionalitäten und Vorgehensweisen dar.
- entwerfen einfache dynamische Datenstrukturen im Kontext einer objektorientierten Programmiersprache.
- erläutern grundlegende Algorithmen, die auf den vermittelten Datenstrukturen arbeiten.
- entwerfen für Programme mittlerer Komplexität durch Einsatz geeigneter Elemente der Programmiersprache Java eine angemessene Modularisierung und legen entsprechende Schnittstellen zwischen den Modulen fest.
- benennen die Grundregeln der benutzungsgerechten Gestaltung von Programmen und nutzen diese, um Benutzungsoberflächen von Programmen begrenzter Funktionalität sowohl strukturell als auch funktional angemessen zu gestalten.
- kennen die grundlegenden Klassen und ihre Operationen, mit denen dateibezogene Operationen implementiert werden können.

Die Studierenden ...

- kennen die Basiskonzepte objektorientierter Programmiersprachen und können sie in Java umsetzen.
- können einfache dynamische Datenstrukturen im Kontext einer objektorientierten Programmiersprache umsetzen und grundlegende Algorithmen auf diesen Datenstrukturen anwenden.
- sind firm in Nutzung einer aktuellen Version einer verbreiteten Entwicklungsumgebung (IntelliJ).
- können ein vollständiges Software-System kleineren Umfangs ausgehend von einer verbalen Aufgabenstellung realisieren.
- entwickeln Software erfolgreich im kleinen Team.
- ermitteln geeignete Testfälle zur Qualitätssicherung.
- kennen die Grundregeln zur Gestaltung benutzungsgerechter Oberflächen und bedienfreundlicher Software.

Voraussetzungen und Empfehlungen:

Die im intoCoding-Bootcamp vermittelten Konzepte sollten verstanden sein und flüssig umgesetzt werden können. Die Installation der Entwicklungsumgebung IntelliJ auf dem eigenen Rechner ist empfehlenswert.

Literatur:

- Christian Ullenboom: Java ist auch eine Insel. 17. Auflage, Rheinwerk Verlag, 2023
- Hans-Peter Habelitz: Programmieren lernen mit Java. 7. Auflage, Rheinwerk Computing, 2022

- Michael Bonacina: Java Programmieren für Einsteiger: Der leichte Weg zum Java-Experten! 2. Auflage, BMU Verlag, 2018
- Markus Neumann: Java Kompendium: Professionell Java programmieren lernen. BMU Verlag, 2019
- Dietmar Ratz et al.: Grundkurs Programmieren in Java. 8. Auflage, Carl Hanser Verlag, 2018
- Michael Inden: Einfach Java: Gleich richtig programmieren lernen. dpunkt.verlag, 2021
- David Kopec: Algorithmen in Java, 32 Klassiker vom Rucksackproblem bis zu neuronalen Netzen, 1. Aufl. Rheinwerk Computing, 2021
- Kathy Sierra et al.: Java von Kopf bis Fuß: Eine abwechslungsreiche Entdeckungsreise durch die objektorientierte Programmierung. O'Reilly, 2023
- Ralph Steyer: Einführung in JavaFX/OpenJFX: Moderne GUIs für RIAs und Java-Applikationen. 2. Aufl., Springer Vieweg, 2022
- Anton Epple: JavaFX 8: Grundlagen und fortgeschrittene Techniken. dpunkt.verlag, 2015
- Sergey Grinev: Mastering JavaFX 10: Build advanced and visually stunning Java applications. Packt Publishing, 2018
- Herbert Schildt: Introducing JavaFX 8 Programming (Oracle Press). Mcgraw-Hill Education, 2015

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB358 – Study Bootcamp Informatik

Verantwortliche:	Dennis Proppe
Moduldauer:	2 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB352 – Study Bootcamp Informatik
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Portfolio-Prüfung
Prüfungsdauer:	60 Min.
ECTS:	15,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	32 Semesterwochenstunden
Termine im Semester:	7 Termine
Zeit in Veranstaltungen:	140 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	90 Stunden
Aufwand während Semesterferien:	40 Stunden
Flexibel einteilbarer Aufwand:	200 Stunden
Gesamtaufwand:	470 Stunden
Lehrende:	Dennis Säring Lars Neumann Malte Heins Dennis Proppe

Studieninhalte:

- Imperative Programmierung: Dies ist der Kern des Moduls. Systematische Einführung in das algorithmische Denken und die strukturierte Problemlösung. Vermittlung der fundamentalen Konzepte der Programmierung anhand einer modernen Programmiersprache, inklusive EVA-Prinzip, Variablen, elementare und komplexe Datentypen, Kontrollstrukturen (Sequenz, Selektion, Iteration), Funktionen und Modularisierung. Entwicklung von Algorithmen und einfachen Datenstrukturen und deren Darstellung z.B. mittels Struktogrammen.
- Grundlagen der Systembedienung und Shell-Programmierung (Unix): Einführung in die Arbeit auf der Kommandozeile. Vermittlung der Navigation im Dateisystem, der Verwaltung von Dateien und Prozessen sowie der Benutzerrechte. Grundlagen der Shell-Programmierung zur Automatisierung von Aufgaben mittels Skripten, Pipes und Umleitungen.
- Grundlagen der Informationstechnik und der Informationstheorie: Vermittlung eines fundamentalen Verständnisses für den Aufbau und die Funktionsweise digitaler Systeme. Dies umfasst die Geschichte der Informationstechnik, die Von-Neumann-Architektur, die Kernkomponenten eines Rechners (CPU, Speicher, Ein-/Ausgabe), Grundlagen von Schaltnetzen sowie die theoretischen Grundlagen der Datenrepräsentation (Binär- und Hexadezimalsystem, Zweierkomplement).
- Praktische Programmierprojekte: Kontinuierliche Anwendung und Vertiefung der theoretischen Inhalte durch zahlreiche praktische Übungen, Programmieraufgaben, begleitende Programmiertests

Lernergebnisse:

Nach erfolgreichem Abschluss des Moduls "Study Bootcamp Programmieren" besitzen die Studierenden folgende Kompetenzen:

Fachkompetenzen:

- Systembedienung: Einen Computer sicher und effizient über die Kommandozeile zu bedienen und einfache Abläufe durch Shell-Skripte zu automatisieren.
- Algorithmisches Denken: Problemstellungen zu analysieren, in logische Teilschritte zu zerlegen und dafür systematisch Algorithmen zu entwickeln.
- Programmierung: Die grundlegenden Konzepte der imperativen Programmierung zu verstehen und diese praktisch in einer modernen Programmiersprache umzusetzen, um lauffähige

Programme zu erstellen.

- Code-Qualität: Strukturierten, lesbaren und kommentierten Code zu verfassen und grundlegende Debugging-Strategien anzuwenden.
- IT-Architektur: Den prinzipiellen Aufbau eines Computers und die grundlegende Funktionsweise von Rechnernetzen zu erläutern.
- Zahlensysteme: Sicher zwischen verschiedenen Zahlensystemen (Dezimal, Binär, Hexadezimal) umzurechnen und die binäre Repräsentation von Daten zu verstehen.

Methodenkompetenzen:

- Problemlösung: Komplexe Aufgabenstellungen systematisch zu erfassen und in eine programmiertechnische Lösung zu überführen.
- Abstraktionsvermögen: Konkrete Anforderungen in abstrakte Datenstrukturen und logische Abläufe zu übersetzen.
- Selbstgesteuertes Lernen: Eigenständig Problemlösungen zu recherchieren und sich mithilfe von Dokumentationen und anderen Ressourcen in neue technische Sachverhalte einzuarbeiten.
- Fehleranalyse: Systematisch Fehler im eigenen Code zu identifizieren, zu analysieren und zu beheben (Debugging).

Sozial- und Personale Kompetenzen:

- Teamarbeit: In kleinen Teams (z.B. via Pair Programming) effektiv zusammenzuarbeiten, um gemeinsam Programmieraufgaben zu lösen.
- Kommunikation: Eigene Lösungsansätze und Code-Strukturen verständlich zu erklären und technische Sachverhalte präzise zu diskutieren.
- Selbstorganisation & Zeitmanagement: Die Bearbeitung von Programmieraufgaben und Projekten eigenverantwortlich zu planen und Meilensteine termingerecht zu erreichen.
- Frustrationstoleranz: Resilienz und Ausdauer bei der systematischen und oft langwierigen Suche nach Fehlern in Software zu zeigen.

Voraussetzungen und Empfehlungen:

Interesse für Informatik, Computer und Softwareentwicklung.

Literatur:

Für Informationstechnik:

Gumm, Hans-Peter; Sommer, Manfred: Einführung in die Informatik, Oldenbourg, 8. Auflage 2009.

Müller, Käser, et., al.: Technische Informatik 1, vdf-Hochschulverlag Zürich, 2003

Schiffmann, Schmitz: Technische Informatik 2, Grundlagen der Computertechnik, Springer-Verlag 1998

Martin: Einführung in die Rechnerarchitektur, Fachbuchverlag Leipzig, 2003

Für Go:

Bodner, J. (2021). Learning Go: An idiomatic approach to real-world Go programming. O'Reilly Media.

McGavren, J. (2019). Head First Go: A brain-friendly guide (1st ed.). O'Reilly Media.

Harsanyi, T. (2022). 100 Go mistakes and how to avoid them. Manning Publications Co.

Für Unix:

Kofler, Michael: Linux: Das umfassende Handbuch, Rheinwerk Computing, 2023, ISBN: 978-3836284424

Dietze, Martin: Praxiskurs Unix-Shell, O'Reilly Verlag GmbH & Co. KG; 2011, ISBN: 978-3897215658

Robbins, Arnold; Beebe, Nelson H.F.: Klassische Shell-Programmierung, O'Reilly Verlag GmbH & Co. KG; 2006, ISBN: 978-3897214415

Kernighan, Brian W.; Pike, Rob: UNIX-Werkzeugkasten: Programmieren mit UNIX, Hanser Fachbuch, 1986, ISBN: 978-3446142732

Friedl, Jeffrey E. F.: Reguläre Ausdrücke, O'Reilly Verlag GmbH & Co. KG, 2007, ISBN: 978-3897217201

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB377 – Deskriptive Statistik und Lineare Algebra

Verantwortliche:	Franziska Bönte Andreas Haase
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB373 – Deskriptive Statistik, Lineare Algebra
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	5.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 135 Stunden weicht signifikant von erwarteten 5.0 ECTS ab.</i>
Lehrende:	Franziska Bönte Andreas Haase

Studieninhalte:

Im Rahmen der linearen Algebra werden folgende Themen behandelt:

- Vektorrechnung
 - Geometrische und arithmetische Vektoren
 - Rechenregeln
 - Lineare (Un-)Abhängigkeit
- Lineare algebraische Gleichungssysteme
 - Gauß-Algorithmus
 - Systematisierung des Lösungsverhaltens
 - Unterbestimmte Systeme
- Matrixrechnung
 - Matrixalgebra
 - Inverse Matrix
 - Matrixgleichungen
 - Zusammenhang mit linearen Gleichungssystemen
 - Rang einer Matrix, Rangkriterium
- Determinanten
 - Definition
 - Zusammenhang mit linearen Gleichungssystemen

Im Rahmen der beschreibenden / deskriptiven Statistik werden folgende Themen behandelt:

- Begrifflichkeiten
- Lage- und Streuungsmaße
- Abhängigkeitsmessung bei qualitativen, komperativen und quantitativen Merkmalen insbesondere Regressionsanalyse
- Deskriptive Zeitreihenanalyse mit Trend-, Saison- und Restkomponentenschätzung nach unterschiedlichen Methoden
- Meß- und Indexzahlen

Lernergebnisse:

Nach der Lehrveranstaltung können die Studierenden ...

- Statistische Daten verdichten und graphisch aussagekräftig darstellen
- Wesentliche Aussagen über Daten anhand geeigneter Kennzahlen treffen und interpretieren
- Die Ableitung von Regressionsformeln verstehen und komplexe Regressions- und deskriptive Zeitreihenanalysen abgestimmt auf den jeweiligen Datensatz durchführen und interpretieren
- sicher im Umgang mit Meß- und Indexzahlen agieren

Nach dem erfolgreichen Besuch der Vorlesung sind die Lernenden in der Lage ...

- lineare algebraische Gleichungssysteme mittels des Gauß-Algorithmus in die Lösbarkeitskategorien (eindeutig lösbar, unendlich viele Lösungen, unlösbar) einzuteilen und ggfs. die Lösung anzugeben.
- die Techniken und Methoden der Vektorrechnung anzuwenden.
- die Techniken und Methoden der Matrixrechnung anzuwenden.
- die Determinante einer niedrigdimensionalen Matrix zu berechnen und den Zusammenhang der Determinante zur Lösungstheorie linearer Gleichungssysteme herzustellen
- einfache technische oder ökonomische Systeme mittels der Techniken und Methoden der linearen Algebra zu modellieren und aus der ermittelten Lösung der mathematischen Formulierung das System quantitativ zu beurteilen.

Voraussetzungen und Empfehlungen:

Grundlegende mathematische Kenntnisse, wie sie im Mathematik-Brückenkurs vermittelt werden, werden vorausgesetzt.

Literatur:

- PAPULA, Lothar:
Mathematik für Ingenieure und Naturwissenschaftler,
Band 2, Teil I. 15. Aufl. Wiesbaden: Springer Vieweg Verlag 2025
- HELM, Werner; PFEIFER, Andreas; OHSER, Joachim:
Mathematik für Wirtschaftswissenschaftler.
1. Aufl. München: Carl Hanser Verlag 2011
- GRAMLICH, Günter:
Lineare Algebra: Eine Einführung.
5. Aufl. München: Carl Hanser Verlag 2021
- TESCHL, Gerald; TESCHL, Susanne:
Mathematik für Informatiker,
Band 1: Diskrete Mathematik und lineare Algebra.
4. Aufl. Heidelberg: Springer Verlag 2013
- FISCHER, Gerd:
Lineare Algebra: Eine Einführung für Studienanfänger.
20. aktualisierte Aufl. Wiesbaden: Springer Verlag 2025
- Christensen, B.; Christensen, S.; Missong, M.: Statistik klipp & klar; 2019; Springer Gabler Verlag
- Bamberg, G.; Baur, F; Krapp, M: Statistik; 18. Auflage; 2017; De Gruyter Oldenbourg Verlag; München
- Missong, Martin; Aufgabensammlung zur deskriptiven Statistik; 2005; 7. Auflage; Verlag R. Oldenbourg, München.
- Schneider, Wolfgang; Kornrumpf, J.; Mohr, Walter; Statistische Methodenlehre --- Definitions- und Formelsammlung zur deskriptiven und induktiven Statistik mit Erläuterungen; 1993; Verlag Oldenbourg, München.

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science

- Betriebswirtschaftslehre Bachelor of Science
 - Computer Games Technology Bachelor of Science
 - Data Science & Artificial Intelligence Bachelor of Science
 - E-Commerce Bachelor of Science
 - Informatik Bachelor of Science
 - IT-Ingenieurwesen Bachelor of Science
 - IT-Management & Consulting Bachelor of Science
 - Marketing, Media & AI Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
 - Wirtschaftsingenieurwesen Bachelor of Science
-

◆ MB032 – Übertragungstechnik

Verantwortliche:	Carsten Burmeister
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch/englisch

Bestandteile:

Teilleistung:	TB182 – Übertragungstechnik
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	90 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	48 Stunden
Aufwand während Semesterferien:	72 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Carsten Burmeister

Studieninhalte:

- Signalformen: Signalanalyse und -synthese
 - Sinusförmige Signale
 - Nicht-sinusförmige periodische Signale
 - Nicht-periodische Signale
- Komplexe Wechselstromrechnung
 - Zeigerdarstellung von sinusförmigen Größen
 - Wechselstromkreise mit Widerstand, Kondensator und Spule
- Ersatzschaltbilder realer elektrischer Bauteile
- Leistungsberechnung im Wechselstromkreis
- Filterstrukturen
 - Tiefpass und Hochpass 1. Ordnung
 - Tiefpass und Hochpass 2. und höherer Ordnung
 - Bandpass und Bandsperre 2. und höherer Ordnung
- Schwingkreise
 - Reihenschwingkreis
 - Parallelschwingkreis
 - Frequenzgänge, Ortskurven und Bodediagramm

Lernergebnisse:

Die Studierenden ...

- besitzen ein Verständnis für Signalformen, -verzerrungen und -verarbeitung bei der Übertragung analoger und diskreter Signale.
- können elektrische Schaltungen in der Nachrichtenübertragung anwenden.
- besitzen die Kenntnis hinsichtlich der Maßnahmen zur Qualitätssicherung bei der Signalübertragung.
- kennen physikalische und logische Übertragungsnetzstrukturen.

Voraussetzungen und Empfehlungen:

- Grundlagen der Elektrotechnik: Sicherer Umgang mit den Grundbegriffen der Elektrotechnik (Strom, Spannung, Widerstand, Leistung) und den grundlegenden Gesetzen (Ohmsches Gesetz,

Kirchhoffsche Regeln).

- Gleichstromnetzwerke: Verständnis für den Aufbau und die Funktionsweise von Gleichstromnetzwerken sowie die Fähigkeit, einfache Schaltungen zu analysieren.

Literatur:

- Hagmann, G.: Grundlagen der Elektrotechnik. Aula-Verlag, 2000 (7. Auflage)
- Meyer, M.: Kommunikationstechnik. Vieweg-Teubner, 2008 (3. Auflage)
- Paul, S.: Grundlagen der Elektrotechnik und Elektronik 2: Elektromagnetische Felder und ihre Anwendungen. Springer-Verlag, 2019 (2. Auflage)
- Kasikci, I.: Wechselstromschaltungen. Springer Vieweg Berlin, Heidelberg, 2025 (5. Auflage)

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB040 – Algorithmen und Datenstrukturen

Verantwortliche:	Christian Uhlig
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB015 – Algorithmen und Datenstrukturen
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	36 Stunden
Aufwand während Semesterferien:	24 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Christian Uhlig

Teilleistung:	TB016 – Übg. Algorithmen und Datenstrukturen
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	75 Min.
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	57 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 72 Stunden weicht signifikant von erwarteten 2.0 ECTS ab.</i>
Lehrende:	Malte Heins

Studieninhalte:

- Analyse von Algorithmen
 - Laufzeit und Speicherbedarf
 - Groß-O / Groß-Omega / Groß-Theta Notationen
 - Amortisierte Laufzeitanalyse
 - Iterative vs rekursive Implementierungen
- Sortieren und Suchen
- Listenstrukturen
 - Verkettete Listen (lineare Listen, Ringlisten, einfach und doppelt verkettete Listen)
 - Arraybasierte Listen
 - Skiplisten
- Baumstrukturen
 - Binäre Suchbäume
 - Balancierte Suchbäume: 2-3-Bäume
 - Balancierte Binäre Suchbäume: Rot/Schwarz-Bäume
 - Spreizbäume
 - Tries
 - Arraybasierte Binäre Heaps
- Hash-Tabellen
- Abstrakte Datentypen und ihre Implementierung
 - Listen
 - Mengen
 - Verzeichnisse
 - Warteschlangen
- Java Collections Framework

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden im Rahmen der Übungsaufgaben praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache Java behandelt, die nicht Bestandteil der Vorlesung sind.

Lernergebnisse:

Die Studierenden ...

- analysieren, diskutieren und vergleichen einfache Algorithmen und Datenstrukturen hinsichtlich ihres Bedarfs an Laufzeit und Speicher.
- differenzieren bei der Analyse von Algorithmen hinsichtlich best case, worst case und average case.
- differenzieren die Laufzeit von Algorithmen nach ihrem konstanten Faktor und ihrem Wachstum in Abhängigkeit von der Problemgröße.
- beurteilen die Laufzeit von Algorithmen ausgehend von Komplexitätsklassen in den Groß-O-, Groß-Omega- und Groß-Theta-Notationen.
- nennen und erläutern wesentliche Aspekte, Funktionsweisen und Eigenschaften von Algorithmen zum Suchen und Sortieren.
- erläutern die Differenzierung in abstrakte Datentypen und ihre Implementierung.
- nennen und erläutern typische abstrakte Datentypen wie Listen, Mengen, Verzeichnisse und Warteschlangen mit ihren Operationen und Anwendungsbereichen.
- nennen und erläutern Motivation, Funktionsweise und Eigenschaften typischer Implementierungen abstrakter Datentypen mit verketteten Listen, Arrays, Baumstrukturen und Hash-Tabellen.
- wählen zu einer gegebenen Problemstellung einen geeigneten abstrakten Datentypen nebst einer geeigneten Implementierung aus.
- wenden die Elemente allgemein der objektorientierten Programmierung und speziell der Programmiersprache Java zur Lösung algorithmischer Problemstellungen an
- wenden die abstrakten Datentypen und Implementierungen des Java Collections Frameworks an

Voraussetzungen und Empfehlungen:

Vorausgesetzt werden Grundkenntnisse der prozeduralen und der objektorientierten Programmierung, insbesondere in der Programmiersprache Java. Diese Kenntnisse sollten insbesondere die Abbildung abstrakter Datentypen per Interfaces und abstrakter Klassen und die Verwendung einfacher generischer Typen umfassen. Es empfiehlt sich, bereits vorhandenes Grundlagenwissen zu Arraylisten, zu verketteten Listen und zu Sortialgorithmen im Vorwege aufzufrischen.

Literatur:

- Sedgewick, Robert; Wayne, Kevin: Algorithms, 4th Edition, Addison-Wesley, 2011
- Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford: Introduction to Algorithms, 3rd Edition, The MIT Press, 2009
- Knuth, Donald E.: The Art of Computer Programming Vol. 1 Fundamental Algorithms, 3rd Edition, Addison-Wesley, 1997
- Knuth, Donald E.: The Art of Computer Programming Vol. 3 Sorting and Searching, 2nd Edition, Addison-Wesley, 1998
- Wirth, Niklaus: Algorithmen und Datenstrukturen, 5. Auflage, Teubner, 2013
- Aho, Alfred V.; Hopcroft, John E.; Ullman, Jeffrey D.: The Design and Analysis of Computer Algorithms, 1st Edition, Pearson, 1975
- Aho, Alfred V.; Hopcroft, John E.; Ullman, Jeffrey D.: Data Structures and Algorithms, Addison-Wesley, 1983
- Aho, Alfred V.; Ullman, Jeffrey D.: Foundations of computer science, Computer Science Press, 1992
- Dokumentation zur Java-API

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science

- IT-Management & Consulting Bachelor of Science
 - IT-Management / -Consulting & -Auditing Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
-

◆ MB043 – Systemnahe Programmierung

Verantwortliche:	Christian Uhlig
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB072 – Systemnahe Programmierung
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	2,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	25 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Christian Uhlig

Teilleistung:	TB074 – Übg. Systemnahe Programmierung
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	35 Min.
ECTS:	3,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	87 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 102 Stunden weicht signifikant von erwarteten 3,0 ECTS ab.</i>
Lehrende:	Malte Heins

Studieninhalte:

- Typische Elemente und Eigenschaften eines C-Programms
- Datentypen
 - Ganzzahl- und Aufzählungstypen, Wahrheitswerte als Ganzzahlen
 - Fließkommatypen, Grundlagen von Fließkommazahlen
 - Strukturierte Typen
 - Vereinigungstypen
 - Zeigertypen
 - Arraytypen
- Funktionszeiger und ihre Anwendungsbereiche
- Konvertierungen
- Arrays und ihre Beziehung zu Zeigern
- Ausdrücke
 - Konstanten
 - Grundlegende Ausdrücke (Zuweisungen, Funktionsaufrufe, etc.)
 - Arithmetische Ausdrücke
 - Boolesche Ausdrücke, Vergleichsoperatoren, logische Operatoren
 - Bitweise Operatoren
 - Arbeit mit Zeigern und Zeigerarithmetik
 - Vorrang und Assoziativität
 - Aspekte der Auswertung (Auswertungsreihenfolge, verkürzte Auswertung, sequence points)
- Anweisungen, insbesondere Verzweigungen und Schleifen
- Dynamische Speicherverwaltung
- Übersetzungsprozess und C-Präprozessor
- Funktionsaufrufe in Maschinen, Aufrufstapel
- Gefahren der Sprache C am Beispiel eines Buffer Overflows mit Manipulation der Rücksprungadresse

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden im Rahmen der Übungsaufgaben praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache C und der C-Standardbibliothek behandelt, die nicht Bestandteil der Vorlesung sind.

Lernergebnisse:

Die Studierenden ...

- formulieren Programme in der Programmiersprache C unter Berücksichtigung der Besonderheiten der Programmiersprache insbesondere in Hinblick auf undefiniertes Verhalten, Plattformabhängigkeiten und Unsicherheiten bestimmter Sprachkonstrukte (z.B. Zeigerarithmetik und fehlende Boundary Checks).
- erläutern in groben Zügen die Repräsentation von Daten und die Abläufe in einem Rechner bei der Ausführung von Anweisungen und Auswertung von Ausdrücken in einer höheren Programmiersprache, insbesondere im Rahmen von Unterprogrammaufrufen.
- erstellen maschinennahe Programme unter besonderer Berücksichtigung von Effizienzaspekten bezogen auf den konstanten Faktor des realisierten Algorithmus
- erläutern typische Gefahren bei Verwendung der Programmiersprache C wie z.B. buffer overflows und berücksichtigen diese Aspekte in der Softwareentwicklung

Voraussetzungen und Empfehlungen:

Vorausgesetzt werden Grundkenntnisse in statisch getypten imperativen Programmiersprachen, die insbesondere charakteristische Datentypen und Kontrollstrukturen (Sequenz, Selektion, Iteration) umfassen und idealerweise auch bereits den Umgang mit Zeigern. Diese Kenntnisse sollten mit gefestigter Programmierpraxis in einer entsprechenden Sprache verbunden sein. Gegebenenfalls empfiehlt es sich, die Kenntnisse am Beispiel einfacher Programmieraufgaben im Vorwege aufzufrischen, um den Einstieg zu erleichtern.

Weiterhin wird ein sicherer Umgang mit der Kommandozeile zum Einsatz der Softwarewerkzeuge in der Übung vorausgesetzt. Gegebenenfalls empfiehlt es sich, die entsprechenden Kenntnisse vorzugsweise am Beispiel der UNIX-Kommandozeile im Vorwege aufzufrischen.

Literatur:

- Harbison, Samuel; Steele, Guy L.: C - A Reference Manual, 5th edition, Prentice Hall, New Jersey, 2002
- Kernighan, Brian W.; Ritchie, Dennis M.: C Programming Language, Prentice Hall, New Jersey, 1998
- Standard zur Programmiersprache, insbesondere ISO/IEC 9899:1999 und ISO/IEC 9899:2011

Studiengänge:

- Computer Games Technology Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB046 – Ingenieurmathematik

Verantwortliche:	Dominik Miller
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch/englisch

Bestandteile:

Teilleistung:	TB165 – Ingenieurmathematik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	120 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Dominik Miller

Studieninhalte:

Teil 1: Höhere Analysis

1. Funktionen mehrerer Variablen
 1. Abhängigkeiten, Mengen und Definitionen
 2. Darstellungen
 3. Grenzwert und Stetigkeit
2. Differentialrechnung für Funktionen mehrerer Variablen
 1. Partielle Ableitung
 2. Verallgemeinerung der Kettenregel
 3. Implizite Darstellung
 4. Relative (lokale) Extremwerte
 5. Extremwerte unter Nebenbedingungen
3. Integralrechnung
 1. Linienintegrale
 2. Doppelintegrale
 3. Dreifachintegrale
4. Gewöhnliche Differentialgleichungen 1. und 2. Ordnung
 1. Gewöhnliche Differentialgleichungen
 2. Differentialgleichungen 1. Ordnung
 3. Lineare Differentialgleichungen 1. Ordnung
 4. Lineare Differentialgleichungen 2. Ordnung mit konstanten Koeffizienten
 5. Weitere Differentialgleichungen

Teil 2: Numerische Mathematik

1. Rechnerarithmetik
 1. Gleitkommazahlen
 2. Fehlerberechnung
2. Numerische Lösung von Nullstellenproblemen
 1. Bisektionsverfahren
 2. Fixpunktiteration
 3. Newton-Verfahren
 4. Sekantenverfahren
 5. Konvergenz
3. Numerische Lösung linearer Gleichungssysteme

1. Gauß-Algorithmus und Dreieckszerlegung
2. Iterative Verfahren
3. Fehlerrechnung
4. Interpolation & Splines
 1. Interpolation, Polynome und kubische Splines
 2. Approximation & lineare Anpassungsrechnung
5. Numerische Differentiation und Integration
 1. Numerische Differentiation
 2. Numerische Integration
6. Anfangswertprobleme gewöhnlicher Differentialgleichungen
 1. Euler-Verfahren
 2. Mittelpunktsregel
 3. Verfahren von Heun

Lernergebnisse:

Die Veranstaltung gliedert sich in zwei sukzessive Teile.

Teil 1: Höhere Analysis.

Die Lernenden können nach dem erfolgreichen Besuch ...

- eine skalare Funktion von mehreren Variablen einmal und mehrfach nach allen Variablen ableiten.
- das totale Differenzial einer mehrdimensionalen skalaren Funktion bilden und seine Bedeutung erklären.
- die mehrdimensionale Kettenregel und die implizite Differenziation anwenden.
- die Lage der lokalen Extrema einer mehrdimensionalen skalaren Funktion, mit und ohne Nebenbedingung, berechnen.
- Flächen und Volumenintegrale berechnen.
- ausgewählte Klassen gewöhnlicher Differenzialgleichungen erster und zweiter Ordnung nach Lösungsmethode klassifizieren und mittels der vorgestellten Verfahren lösen.

Teil 2: Numerische Mathematik

Die Lernenden können nach dem erfolgreichen Besuch ...

- die Notwendigkeit für numerische Verfahren anführen.
- die prinzipiellen Beschränkungen und Fehler numerischer Verfahren aufzählen und darlegen.
- Nullstellen von skalaren nichtlinearen Funktionen mittels der vorgestellten Methoden näherungsweise bestimmen und die Güte der Approximation mittels Fehleranalyse untersuchen.
- lineare Gleichungssysteme numerisch mittels direkter und iterativer Verfahren lösen und die Güte des erhaltenen Ergebnisses mittels Fehleranalyse evaluieren.
- eine gegebene Menge von Datenpunkten interpolieren. Insbesondere können die Lernenden das einfache Interpolationspolynom berechnen und sind in der Lage eine lineare stückweise Interpolierende zu berechnen.
- eine gegebene Menge von Datenpunkten mittels einer Menge von Ansatzfunktionen approximieren. Dabei können sie das zu Grunde liegende Minimierungsproblem selbständig formulieren und lösen.
- eine gegebene eindimensionale Funktion numerisch differenzieren und integrieren und die Fehler der Algorithmen bewerten und die Fehler des Ergebnisses berechnen.
- eine gegebene gewöhnliche Differenzialgleichung erster Ordnung mittels verschiedener Einschrittverfahren näherungsweise lösen und den Fehler des Ergebnisses unter Verwendung der Fehleranalyse abschätzen.
- Programmiererfahrene Lernende können die dargestellten Algorithmen in entsprechende Computercodes übersetzen.

Voraussetzungen und Empfehlungen:

- Analysis und Lineare Algebra

Literatur:

- PAPULA, Lothar: Mathematik für Ingenieure und Naturwissenschaftler. Band 2, 13. durchgesehene Aufl. Wiesbaden: Vieweg + Teubner 2012
- PAPULA, Lothar: Mathematik für Ingenieure und Naturwissenschaftler. Band 3, 6. überarbeitete und erweiterte Aufl. Wiesbaden: Vieweg + Teubner 2011
- KNORRENSCHILD, Michael: Numerische Mathematik: Eine beispielorientierte Einführung, 5. aktualisierte Aufl. München: Carl Hanser Verlag 2013
- SCHWARZ, Rudolf; KÖCKLER, Norbert: Numerische Mathematik. 8. aktualisierte Aufl. Wiesbaden: Vieweg + Teubner 2011
- Opfer, Gerhard – Numerische Mathematik für Anfänger. 5. aktualisierte Aufl. (2008), Vieweg & Teubner
- Press, William; Teukolsky, Saul; Vetterling, William und Flannery, Brian – Numerical Recipes. 1986, Cambridge University Press

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB048 – Elektronik

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB185 – Elektronik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	8 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	25 Stunden
Aufwand während Semesterferien:	2 Stunden
Flexibel einteilbarer Aufwand:	3 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 60 Stunden weicht signifikant von erwarteten 5,0 ECTS ab.</i>
Lehrende:	Sergei Sawitzki

Studieninhalte:

- Passive Bauelemente und Schalter
 - Übersicht der elektronischen Bauelemente, E-Normreihen
 - Widerstand, Kondensator, Spule
 - Schalter
- Thermisches Verhalten von Bauelementen
 - Einleitung und Grundbegriffe, Wärmestromkreis
 - Aspekte des thermischen Verhaltens
 - Erwärmung und Abkühlung, Einsatz von Kühlkörpern
- Lineare Netzwerke bei Gleichstrom
 - Strom- und Spannungsquellen, Grundstromkreis
 - Bestimmung des Arbeitspunktes
 - Allgemeine Netzwerkanalyse
- Lineare Netzwerke bei zeitabhängiger Erregung
 - Wechselstrom und Wechselspannung
 - Passive Filter, Schwingkreise und Resonatoren
 - Wechselstrombrücken
 - Ausgleichsvorgänge
- Anwendungen der Systemanalyse
 - Einleitung
 - Fourier-Reihen, Klirrfaktor
 - Fourier- und Laplace-Transformation
 - Vierpoltheorie

Lernergebnisse:

Die Studierenden ...

- kennen die theoretischen Grundlagen der Elektronik
- kennen Methoden und Werkzeuge für Entwurf und Analyse von elektronischen Systemen
- können das reale Verhalten von elektronischen Bauteilen (Toleranzen, Temperaturabhängigkeiten usw.) beim Schaltungsentwurf berücksichtigen
- können einfache analoge und digitale Systeme aus dem Bereich der technischen Informatik begreifen, spezifizieren und entwerfen

- können mit modernen Entwurfs- und Simulationswerkzeugen arbeiten

Voraussetzungen und Empfehlungen:

Kenntnisse der elektrotechnischen Grundlagen, Umgang mit komplexen Zahlen und Funktionen, Infinitesimalrechnung

Literatur:

- Tietze, Ulrich; Schenk, Christoph: Halbleiterschaltungstechnik, 16. Auflage Springer Verlag, 2016
- Stiny, Leonhard: Handbuch passiver elektronischer Bauelemente, Franzis Verlag 2007
- Hering, Ekbert; Bessler, Klaus; Gutekunst, Jürgen: Elektronik für Ingenieure und Naturwissenschaftler, Springer Verlag 2005
- Lunze, Klaus: Berechnung elektrischer Stromkreise, 15. Auflage Huss Medi 1990
- Horowitz, Paul; Hill, Winfield: Die Hohe Schule der Elektronik. Teil 1: Analogtechnik, 8. Auflage Elektor-Verlag 2006
- Schiffmann, Wolram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Brauer, Harry; Lehmann, Constans; Lindner, Helmut: Taschenbuch der Elektrotechnik und Elektronik, 9. Auflage Hanser Fachbuchverlag, 2008

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB378 – Fortgeschrittene Lineare Algebra

Verantwortliche:	Andreas Haase
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB374 – Fortgeschrittene Lineare Algebra
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Andreas Haase

Studieninhalte:

- Determinanten
 - Definition, Beispiele und Eigenschaften
 - Der Entwicklungssatz von Laplace
 - Cramer'sche Regel und lineare Gleichungssysteme
- Vektorräume
 - Definition, Beispiele und Eigenschaften
 - Unterräume
 - Lineare (Un-)Abhängigkeit, Basis und Dimension
- Euklidische und unitäre Vektorräume
 - Skalarprodukt und Norm
 - Orthogonalität
 - Orthogonal- und Orthonormalbasen
- Analytische Geometrie
 - Darstellung von Geraden und Ebenen
 - Lagebeziehung zwischen linearen geometrischen Objekten
 - Einfache nichtlineare Objekte an Beispielen
- Abbildungen
 - Lineare Abbildungen
 - Affine Abbildungen
 - Koordinatentransformationen
- Eigenwerte und Eigenvektoren
 - Charakteristisches Polynom, Eigenwerte, Eigenvektoren
 - Diagonalisierung von Matrizen
 - Matrixfunktionen

Lernergebnisse:

Nach dem erfolgreichen Besuch der Veranstaltung können die Studierenden ...

- die Determinante einer Matrix beliebiger Dimension berechnen und den Zusammenhang zur Lösungstheorie linearer Gleichungssysteme herstellen.
- die Vektorraumaxiome nennen und eine gegebene Menge mit Verknüpfungen darauf überprüfen, ob diese ein Vektorraum (über $\mathbb{R}$ oder $\mathbb{C}$) ist.

- Die Definition eines Unterraums nennen; Teilmengen von Vektorräumen darauf überprüfen, ob diese Unterräume sind.
- das Konzept der linearen Abhängigkeit von Vektoren erklären; Teilmengen von Vektorräumen auf lineare Abhängigkeit überprüfen.
- die Definition einer Basis nennen. Teilmengen von Vektorräumen darauf überprüfen, ob diese eine Basis sind.
- die Definition eines Skalarproduktes nennen. Verschiedene lineare Abbildungen auf Vektorräume darauf überprüfen, ob diese ein Skalarprodukt sind.
- die Definition einer Norm nennen; den Zusammenhang zwischen Skalarprodukt und Norm nennen.
- die Definition einer Orthonormalbasis nennen; eine Orthonormalbasis aus einer gegebenen Basis konstruieren (Gram-Schmidt-Verfahren).
- die Parameter und Koordinatendarstellung von Geraden und Ebenen formulieren; Lagebeziehungen zwischen linearen geometrischen Objekten berechnen; Lagebeziehungen zwischen linearen und einfachen nichtlinearen Geometrischen Objekten berechnen.
- die Definition einer linearen Abbildung nennen; lineare Abbildungen mittels Matrix-Vektor-Schreibweise ausdrücken. Eigenschaften gegebener linearer Abbildungen bestimmen.
- die Definition einer affinen Abbildung nennen; affine Abbildungen mittels Matrix-Vektor-Schreibweise ausdrücken. Eigenschaften gegebener affiner Abbildungen bestimmen.
- Koordinatentransformationen als affine Abbildung durchführen; die affine Abbildung einer Koordinatentransformation berechnen; aktive und passive Koordinatentransformationen unterscheiden.
- das charakteristische Polynom einer Matrix aufstellen; die Eigenwerte einer Matrix berechnen; die Eigenvektoren einer Matrix berechnen.
- eine Matrix diagonalisieren.
- bestimmte Funktionen einer Matrix berechnen.

Voraussetzungen und Empfehlungen:

Es werden grundlegende Kenntnisse zu Vektoren, Vektoralgebra, Matrizen, Matrixalgebra, Lösen von linearen Gleichungssystemen mittels Gauß-Verfahren vorausgesetzt.

Literatur:

- GRAMLICH, Günter M.:
Lineare Algebra: Eine Einführung.
5. aktualisierte Aufl. München: Carl Hanser Verlag 2021
- FISCHER, Gerd:
Lernbuch Lineare Algebra und Analytische Geometrie.
4. Aufl. Wiesbaden: Vieweg + Teubner Verlag 2019
- ANTON, Howard:
Elementary Linear Algebra.
John Wiley & Sons Inc 2019

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB037 – Rechnernetze

Verantwortliche:	Ilja Kaleck
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB013 – Rechnernetze
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	60 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Ilja Kaleck

Teilleistung:	TB014 – Prakt. Rechnernetze
Lernform:	Praktikum
Prüfungsform:	Abnahme
Prüfungsdauer:	
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Ilja Kaleck

Studieninhalte:

- Allgemeine Grundlagen und Begriffe
 - Allgemeine Strukturen in der Datenkommunikation
 - Protokolle und Protokollabläufe
 - Netztopologien und Klassifizierung von Übertragungsnetzen
- Das ISO-OSI Referenzmodell
 - Prinzip der Schichtenbildung und Schichtenfunktionen im Überblick
 - Datenfluss im Modell
 - Aktuelle Koppellemente zum Netzaufbau im Kontext der OSI-Modells
- Die Internet-Architektur
 - Historie, Architekturübersicht, Standardisierungen
 - IPv4-Adressstrukturen und Netzaufbau, Subnetting
 - UDP-/TCP-Kommunikation, Sockets bzw. Socket-Kommunikation
 - Betrachtung ausgewählter Anwendungsprotokolle (DNS, TELNET / SSH, ...)
 - Network Address Translation (NAT) und der Einsatz von Proxy-Servern (Diensten)
 - Einführung in das neue Internet Protocol Version 6 (IPv6)
 - Adress- und Netzstruktur, Migrationshinweise
 - Änderungen an höheren Protokollen in Bezug auf das IPv6
- Technik Lokaler Netze (LANs)
 - Ablauf der Kommunikation in IEEE 802 LANs (Layer-2, IP, inkl. DHCP)
 - Schwerpunkt Betrachtung: Ethernet-Technik, Zugriffsverfahren und
 - Technische Umsetzungen (10Mbps / 100FE / 1GbE / 10GbE)
 - Überblick über andere LAN-Technologien
- Koppellemente und Vermittlungstechniken
 - Repeater, Brücken- bzw. Layer-2 Switching-Technologie
 - Virtuelle LANs (VLANs), Class-of-Services im LAN
 - Router bzw. IP-Routing, Link-State und Distanzvektor-Verfahren,
 - Hierarchisches Routing und IP-Multicasting
 - Drahtlose Netze nach IEEE 802.11,
 - Struktur, Aufbau, Übertragungskonzepte, Sicherheitsbetrachtungen
- Verzeichnisdienste
 - Einführung und grundlegendes Konzept des X.500
 - Herstellerspezifische Lösungen (Active Directory)
 - Lightweight Directory Access Protocol (LDAP)

Durchführung eines Laborpraktikums durchgängig individuell am eigenen PC-System unter Einsatz dedizierter Wechselfestplatten (einzelner Teilnehmer oder Arbeitsgruppe)

- Einrichtung eines Server-Betriebssystems und Konfiguration der grundlegenden Kommunikationsprotokolle (IPv4, IPv6).
 - Nutzung typischer Internetdienstprogramme und Betrachtung der dabei verwendeten Protokolle.
- Einsatz von Techniken zur Unix/Windows-Integration (NFS, SAMBA, X-Windows, Unix mit Posix-ACLs)
- Nutzung einfacher Benutzer- und Rechteverwaltung im Netz (Domänenkonzept).
- Einsatz von Virtualisierungstechniken auf dem Desktop
 - Aufbau einer lokalen Netzinfrastruktur und Einrichtung des lokalen IP-Routings (inkl. NAT)
 - Grundlegende Firewall-Konfiguration
- Einrichten und Arbeiten mit aktuellen Verzeichnisdiensten
 - Aufbau einer eigenen Verzeichnisstruktur (Directory)
 - Formulierung von Suchanfragen an Verzeichnisdienste (Active Directory, LDAP-Server)
- Konfiguration grundlegender Internet-Serverdienste (DNS, FTP, HTTP, Proxy-Server, TELNET / SSH)
 - Nutzung der SSH Port-Forwarding Funktion
- Protokollanalyse und Fehlersuche im LAN mit einem LAN-Analyzer
 - Nutzung einer Remote-Probes zur verteilten LAN-Analyse im Netz.
 - Einfache LAN-Performance Messungen
- Aufbau einer komplexen virtuellen Unternehmensnetzes
 - Strukturanalyse, Adress- und Routerkonfiguration, Test
 - Einsatz des "CISCO Packet Tracer" Simulations

Lernergebnisse:

Die Studierenden erlangen ...

- ein grundlegendes Verständnis für den Aufbau einer herstellerneutralen Kommunikationsarchitektur (OSI).
- Kenntnisse über den Aufbau und die Funktion des Internet-Architekturmodells.
 - Kenntnis über IPv4-Adress- und Netzstrukturen.
 - Verständnis über die Arbeitsweise essentieller Anwendungsprotokolle.
 - Fähigkeit zum Verständnis des Ablaufs einfacher Interprozesskommunikation, u.a. als Basis für die Realisierung komplexerer verteilter Anwendungen.
 - die Arbeitsweise spezifischer Maßnahmen gegen den IPv4-Adressmangel im IPv4 (NAT, Proxyserver-Dienste) kennen.
 - Wissen über die Eigenschaften des neuen Internet-Protokolls Version 6 (IPv6) und Änderungen an bestehenden Internet-Protokollen (u. a. DNS, ICMP).
- Verständnis über den technischen Aufbau und den Betrieb Lokaler Netze (LANs).
 - Verständnis hinsichtlich des generellen Ablaufs der IP-Kommunikation in LANs.
 - Wissen um die Eigenschaften aktueller Netztechnologien (Schwerpunkt: Ethernet-Technik).
 - Kenntnisse zum Aufbau und Betrieb drahtloser Netze (IEEE 802.11 WLANs).
- Wissen um den technischen Aufbau von Netzstrukturen bzw. des Internets.
 - Wissen um die Aufgabe Funktionsweise der klassischen von Koppelemente in Netzen.
 - elementares Wissen um die Arbeitsweise praxisrelevanter Routingverfahren für kleinere und größere Netze (u. a. einfaches IP-Routing; hierarchisches Routing).
- Grundkenntnisse über den Aufbau und die Funktionsweise von Verzeichnisdiensten.

Die Studierenden erlangen ...

- die Fähigkeit zum praktischen Umgang mit der Internet-Technologie am eigenen PC.
 - die Fähigkeit zum Anschluss von Systemen an ein Unternehmensnetz.
 - die Fähigkeit zur grundlegenden Konfiguration des Internet-Protokolls (IPv4, IPv6).

- das Verständnis für Sicherheitsrichtlinien auf Multi-User Systemen (Windows, Linux).
- die Fähigkeit zur Analyse und Behebung typischer Fehlersituationen im Rahmen der Kommunikation von Anwendungen und Systemen im Netz.
- die Fähigkeit zur Konfiguration grundlegender Internet-Dienste (u. a. DNS, HTTP, FTP).
- das Verständnis für Lösungsansätze aktueller Techniken zur Unix-/Windows Integration in heterogenen Unternehmensnetzen (NFS, SAMBA, X-Windows).
- das Verständnis über aktuelle Konzepte zur Benutzer- und Rechteverwaltung in Netzen.
 - die Fähigkeit zur Benutzerverwaltung mittels eines Domänenkonzeptes (Windows).
 - die Fähigkeit zur Einrichtung von Verzeichnisdiensten (LDAP, Active Directory).
- die Grundkenntnisse zum praktischen Einsatz von Virtualisierungstechniken auf dem Desktop.
 - die Fähigkeit zur Einrichtung einfacher IP-Routingfunktionen auf einem System.
- das Verständnis über den praktischen Aufbau und Betrieb eines WLANs und dessen interne Kommunikationsabläufe (inkl. Sicherheitsbetrachtungen).
- die Fähigkeit zum Einsatz eines LAN-Analyzers zur Analyse von Kommunikationsabläufen zwischen Anwendungen sowie zur Fehleranalyse in LANs und WLANs.
- grundlegende Kenntnisse digitaler Sprachübertragung in Netzen mittels der Voice-over-IP (VoIP) Technik (Wahlthema).
- grundlegende Kenntnisse zu Streaming-Media Technik und den Real-Time Protokollen zur Übertragung multimedialer Inhalte in Netzen (Wahlthema).

Voraussetzungen und Empfehlungen:

Dieses Modul setzt intensive Lesekompetenz voraus. Das Praktikum erfordert aufmerksames Lesen, Befolgen von Aufgabenstellungen und die Fähigkeit zur selbstständigen Arbeit am PC. Das sichere Navigieren in Dateibäumen sowie das Installieren und verwenden von Software primär unter Microsoft Windows wird vorausgesetzt. Ein grundlegendes Verständnis über die haushaltsübliche Nutzung von Netzwerken (WLAN, Internet) wird empfohlen.

Literatur:

- TANNENBAUM, Andrew S.:
Computer Netzwerke.
5. aktualisierte Aufl. München: Pearson Education, 2013, ISBN 978-3-86894-452-5
- KUROSE, James F.; ROSS, Keith W.:
Computer Netzwerke. Der Top-Down Ansatz.
6. Aufl. : Pearson Education, 2014, ISBN 978-3-86894-237-8
- ECKERT, Claudia
IT-Sicherheit: Konzepte – Verfahren – Protokolle
11.. Aufl. : De Gruyter Studium, 2023, ISBN 978-3-11099-689-0
- RECH, Jörg:
Ethernet. Technologien und Protokolle für die Computervernetzung.
3. Aufl. Heidelberg: dPunkt-Verlag, 2014, ISBN 978-3-94409-904-0
- RECH, Jörg:
Wireless LANs. 802.11-WLAN-Technologie und praktische Umsetzung im Detail.
4. Aufl. Heidelberg: dPunkt-Verlag, 2012, ISBN 978-3-936931-75-4
- BADACH, Anatol; HOFFMANN, Erwin:
Technik der IP-Netze. Funktionsweise, Protokolle und Dienste.
2. Aufl. München: Hanser, 2007, ISBN 978-3446215016
- ECKERT, Claudia
IT-Sicherheit: Konzepte – Verfahren – Protokolle
11.. Aufl. : De Gruyter Studium, 2023, ISBN 978-3-11099-689-0
- HEGEMANN, Klaus; SCHAEFER, Udo:
Basiswissen IT-Berufe: Vernetzte IT-Systeme.
9. Aufl.:Bildungsverlag EINS, 2025, ISBN 978-3-8237-1141-4
- BERG, Eric:
Cloud Computing lernen: Netzwerke

<https://de.linkedin.com/learning/cloud-computing-lernen-netzwerke-23895004>

- Aktualisierungsdatum 18.07.2025

- BERG, Eric:

Cloud Computing lernen: Speicher

<https://de.linkedin.com/learning/cloud-computing-lernen-speicher-23486723>

- Aktualisierungsdatum 18.07.2025

-
- RECH, Jörg:

Wireless LANs. 802.11-WLAN-Technologie und praktische Umsetzung im Detail. 4. Aufl. Heidelberg: dPunkt-Verlag, 2012, ISBN 978-3-936931-75-4

- LIU/MATTHEW/PARZIALE/DAVIS/FORRESTER/BRITT:

TCP/IP Tutorial and Technical Overview (PDF). 8th. Ed. 2006: IBM-Redbook Serie.

<http://www.redbooks.ibm.com/redbooks/> Aktualisierungsdatum 29.06.2014

- GROUPE IEEE 802.11: Aktuelle Spezifikationen zu IEEE 802.11.

<http://standards.ieee.org/getieee802/802.11.html> Aktualisierungsdatum 29.06.2014

- IETF: Internet-Draft Dokumente und aktuelle RFCs. <http://www.ietf.org/> - Aktualisierungsdatum 29.06.2014

Studiengänge:

- Computer Games Technology Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB057 – Fortgeschrittene Objektorientierte Programmierung

Verantwortliche:	Christian Uhlig
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB024 – Fortgeschrittene Objektorientierte Programmierung
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	120 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	25 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Christian Uhlig

Teilleistung:	TB025 – Übg. Fortgeschrittene Objektorientierte Programmierung
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	75 Min.
ECTS:	3.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	86 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 101 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Malte Heins

Studieninhalte:

- Generische Typen / Java Generics
- Verschachtelte Typen
- Funktionswerte in OO-Sprachen (Funktionale Interfaces, Lambda-Ausdrücke, Methodenreferenzen)
- Funktionale Programmierung mit Java Streams
- Nebenläufigkeit (Threads, Racing Conditions, Synchronisation, Waitsets, volatile Variablen, Java Memory Model / happens-before-Relation, nebenläufige Auswertung von Streams, Executors, Futures, Fork-Join-Tasks, Thread-sichere Collections)

Bearbeitung von Übungsaufgaben parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen. Zusätzlich werden im Rahmen der Übungsaufgaben praxisrelevante Aspekte der fortgeschrittenen OOP behandelt, die nicht Bestandteil der Vorlesung sind.

Lernergebnisse:

Die Studierenden ...

- wenden fortgeschrittene Konzepte objektorientierter Programmiersprachen wie parametrische Polymorphie und Funktionswerte zur Konstruktion wiederverwendbarer Softwarekomponenten an.
- nennen und erläutern die Grundlagen generischer Datentypen und ihre Korrespondenz mit Konzepten objektorientierter Sprachen (Schnittstellen, abstrakte Klassen, konkrete Klassen, Polymorphie).
- entwerfen und implementieren generische Datentypen mit Java Generics und nutzen hierbei beschränkte Typparameter, generische Methoden und parametrisierte Typen mit Wildcards.

- definieren Funktionswerte per anonymer Klassen, Lambda-Ausdrücke und Methodenreferenzen.
- wenden Prinzipien funktionaler Programmierung in objektorientierten Sprachen am Beispiel von Java Streams an.
- nennen und erläutern Motivation, Grundlagen und Herausforderungen nebenläufiger Programmierung.
- wenden Primitiven nebenläufiger Programmierung in Java an (Erzeugen von Threads, Thread-Kommunikation / Synchronisation, usw.).
- nennen und erläutern die Herausforderungen gemeinsamer Zustände von Threads, das Java-Speichermodell und die happens-before-Relation und wenden diese Kenntnisse zur Vermeidung von racing conditions und data races an.
- nennen und erläutern die Vorzüge funktionaler Programmierung für die parallele Auswertung von Berechnungen am Beispiel von Java Streams.
- definieren Stream-Pipelines unter Berücksichtigung der Anforderungen und Konsequenzen einer nebenläufigen Auswertungsstrategie.

Voraussetzungen und Empfehlungen:

Vorausgesetzt werden gefestigte theoretische und praktische Kenntnisse in objektorientierter Programmierung im Allgemeinen und in der Programmiersprache Java im Speziellen. Gegebenenfalls empfiehlt es sich, die Kenntnisse im Vorwege anhand einfacher Übungsaufgaben aufzufrischen.

Literatur:

- Gosling, James; Joy, Bill; Steele, Guy; Bracha, Gilad; Buckley, Alex; Smith, Daniel: The Java Language Specification, Java SE 11 Edition, 2018
- Bloch, Joshua: Effective Java: A Programming Language Guide, 3rd Edition, Addison-Wesley, 2017
- Lea, Doug: Concurrent Programming in Java: Design Principles and Patterns, 2nd Edition, Addison-Wesley, 1999
- Goetz, Brian; Peierls, Tim; Bloch, Joshua; Bowbeer, Joseph; Holmes, David; Lea, Doug: Java Concurrency in Practice, Addison-Wesley, 2006
- Urma, Raoul-Gabriel; Fusco, Mario; Mycroft, Alan: Java 8 in Action: Lambdas, Streams, and Functional-Style Programming, Manning Publications, 2014
- Ullenboom, Christian: Java ist auch eine Insel, 12. Auflage, Galileo Press GmbH, 2016
- Loudon, Kenneth C.: Programming Languages: Principles and Practice, 2nd Edition, Thomson Learning, 2002

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB068 – Halbleiterschaltungstechnik

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB186 – Halbleiterschaltungstechnik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	65 Stunden
Aufwand während Semesterferien:	5 Stunden
Flexibel einteilbarer Aufwand:	5 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Sergei Sawitzki

Teilleistung:	TB190 – Übg. Elektronik und Halbleiterschaltungstechnik
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	5 Min.
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Timm Bostelmann

Studieninhalte:

- Grundlagen der Halbleiterphysik
- Halbleiter-Bauelemente
 - pn-Übergang, Diode
 - Bipolartransistor
 - Feldeffekttransistoren
 - Überblick über sonstige Bauelemente
 - Grundsaltungen, Verstärker
- Halbleiterschaltungstechnik
 - Strom- und Spannungsquellen
 - Operationsverstärker
 - Transistoren als Schalter, Digitale Schaltungen
- Passive Netze
 - Filterschaltungen
 - Einschwingvorgänge
 - Stabilisierungsschaltungen
- Verstärkerschaltungen
 - Arbeitspunkteinstellungen
 - Statisches und dynamisches Verhalten
 - Gegenkopplungen
- Operationsverstärkerschaltungen
 - Beschaltungen
 - Anwendungen
 - Stabilität

Lernergebnisse:

Die Studierenden ...

- können analoge und digitale elektronische Systeme, soweit sie für die technische Informatik von Belang sind, verstehen und entwerfen
- kennen die Methodik des Schaltungsentwurfs und können diese praktisch einsetzen
- haben ein Verständnis für realitätsnahe Schaltungsentwicklung unter Einbeziehung des realen Bauteilverhaltens (Toleranzen, Streuungen, Temperaturabhängigkeiten usw.)
- können bestehende Schaltungen und Systeme analysieren

- kennen den Aufbau und die Funktionsweise von den wichtigsten Halbleiter-Bauelementen
- erkennen Verbindungen zwischen Digitaltechnik und Halbleiterschaltungstechnik als technologischer Grundlage digitaler Schaltungen und Systeme
- können typische Aufgaben aus dem Stoffumfang der Vorlesungen Elektronik und Halbleiterschaltungstechnik eigenständig lösen
- Kenndaten elektronischer Bauteile aus Datenblättern entnehmen

Voraussetzungen und Empfehlungen:

Erfahrungen im Umgang mit Simulationswerkzeugen (insbesondere Pspice)

Literatur:

- Tietze, Ulrich; Schenk, Christoph: Halbleiterschaltungstechnik, 16. Auflage Springer Verlag, 2016
- Stiny, Leonhard: Handbuch aktiver elektronischer Bauelemente, Franzis Verlag 2009
- Hering, Ekbert; Bessler, Klaus; Gutekunst, Jürgen: Elektronik für Ingenieure und Naturwissenschaftler, Springer Verlag 2005
- Horowitz, Paul; Hill, Winfield: Die Hohe Schule der Elektronik. Teil 1: Analogtechnik, 8. Auflage Elektor-Verlag 2006
- Schiffmann, Wolfram; Schmitz, Robert: Technische Informatik, in 3 Bänden. 3. Auflage Springer Verlag, 1996
- Brauer, Harry; Lehmann, Constans; Lindner, Helmut: Taschenbuch der Elektrotechnik und Elektronik, 9. Auflage Hanser Fachbuchverlag 2008
- Taur, Yuan; Ning, Tak H.: Fundamentals of Modern VLSI Devices, 2nd edition, Cambridge University Press 2009
- Ng, Kwok K.: Complete Guide to Semiconductor Devices, 2nd edition, Jaohn Wiley & Sons 2002

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB073 – Systemtheorie

Verantwortliche:	Carsten Burmeister
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch/englisch

Bestandteile:

Teilleistung:	TB179 – Systemtheorie
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	90 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	48 Stunden
Aufwand während Semesterferien:	72 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Carsten Burmeister

Studieninhalte:

- Beschreibung und Analyse von kontinuierlichen Signalen und Systemen im Zeitbereich
 - Signale und Signalarten
 - Systeme und Systemeigenschaften
 - Impulsantwort und Faltung
- Beschreibung und Analyse von kontinuierlichen Signalen und Systemen im Frequenzbereich
 - Fourieranalyse
 - Fourierreihen
 - Fouriertransformation
- Beschreibung und Analyse von kontinuierlichen Signalen und Systemen im Bildbereich
 - Laplacetransformation
 - Systemstabilität
- Beschreibung und Analyse von zeitdiskreten Signalen und Systemen
 - Abtastung / Abtasttheorem
 - Digitale Signalverarbeitung
 - z-Transformation
 - Diskrete Fouriertransformation und FFT

Lernergebnisse:

Die Studierenden ...

- kennen zeitkontinuierliche und zeitdiskrete Integraltransformationen und deren Eigenschaften.
- können Integraltransformationen zur Spektralanalyse, zur Lösung linearer Differentialgleichungssysteme und zur allgemeinen Analyse nachrichtentechnischer und regelungstechnischer Systeme anwenden.

Voraussetzungen und Empfehlungen:

- Fundierte Kenntnisse in Analysis (Differentiation, Integration, Differentialgleichungen)
- Grundkenntnisse in linearer Algebra (Matrizen, Vektoren, Eigenwerte)
- Kenntnisse in komplexer Analysis sind von Vorteil

Literatur:

Meyer, M.: Signalverarbeitung. Vieweg und Teubner 2011, 6. Auflage

Oppenheim, A., Willsky, S.: Signals and Systems, 2nd Edition. Pearson 1996, 2. Auflage

Chaparro, L.: Signals and Systems using MATLAB. Academic Press 2018, 3. Auflage

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB101 – Echtzeitsysteme

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB063 – Echtzeitsysteme, Interface-Technologie
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	150 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	65 Stunden
Aufwand während Semesterferien:	5 Stunden
Flexibel einteilbarer Aufwand:	5 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Dennis Säring Sergei Sawitzki

Teilleistung:	TB070 – Prakt. Echtzeitsysteme
Lernform:	Praktikum
Prüfungsform:	Abnahme
Prüfungsdauer:	30 Min.
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	4 Termine
Zeit in Veranstaltungen:	5 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	5 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	50 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Timm Bostelmann

Studieninhalte:

- Grundlagen der Prozessdatenverarbeitung (PDV)
- Sensortechniken
- Strategien zur Fusionierung von Prozessdaten
- Prädiktionsmodell Kalman- und Partikel-Filter
- Moderne BUS-Systeme
- Prozesse
 - Grundbegriffe
 - Technische Umsetzung
 - Aufgaben des Betriebssystems
 - Probleme und Lösungsansätze
 - Kommunikationsmechanismen
- Modellierung (Flussdiagramme Petri-Netze)
- Scheduling
 - Einleitung
 - Strategien
 - Zeitverwaltung
 - Beispiele
- Einführungsvorlesung
 - Motivation zur Veranstaltung
 - Beschreibung der Systemumgebung
 - Einführung in die Verwendung des Echtzeitkerns
- Einführungsaufgabe
 - Geführtes Erstellen eines Beispielprojektes
 - Teilweise geführte Programmierung eines Prozesses
 - Selbstständige Programmierung eines Prozesses
- Eine Aufgabe aus dem Umfeld Echtzeit, Multitasking, Simulation wird gestellt und steht den Studierenden als Anforderungskatalog zur Verfügung
 - Struktureller Programmentwurf
 - Kodierung und Test
 - Erstellung einer Dokumentation
 - Abnahme durch den Betreuer

Lernergebnisse:

Die Studierenden ...

- kennen die grundlegenden Begriffe der Prozessdatenverarbeitung
- kennen die mathematischen Modelle für eine prädiktive Zustandsschätzungen auf Basis zurückliegender Messergebnisse
- können die Anforderungen an moderne BUS-Systeme im Kontext von komplexen Kommunikationsstrukturen (z.B. PKW, Flugzeug) einschätzen
- kennen die Anforderungen, die für die Programmierung von Embedded Systems, Systemen mit kleinen Hardwareressourcen und Echtzeitsystemen erforderlich sind
- kennen Modellierungstechniken und -methoden bei Programmierung von nebenläufigen Prozessen, Modellierung und Anwendung von Prozesskommunikations- und Synchronisationsmechanismen
- können Aufgabenstellungen auf Systeme nebenläufiger (kooperierender und konkurrierender) Prozesse abbilden und diese softwaretechnisch umsetzen
- kennen die Methodik der Programmierung paralleler Prozesse, unter besonderer Berücksichtigung von Echtzeitanforderungen
- können einfache Echtzeitsysteme konzipieren und realisieren
- können sich mit Hilfe eines Handbuches in die Schnittstelle eines Echtzeitbetriebssystems einarbeiten
- eine Lösung auf System- und Implementierungsebene in einer schriftlichen Ausarbeitung dokumentieren.

Voraussetzungen und Empfehlungen:

- Grundlegende Kenntnisse von Programmierung und Datenstrukturen
- Kenntnisse der Programmiersprache C
- Kenntnisse von Prozesskommunikations- und Synchronisationsmechanismen
- Grundlegende Kenntnisse von Prozessdatenverarbeitung

Literatur:

- Börcsök: Prozeßrechner- und Automation, Heise-Verlag, 1997
- Jacobsen: Einführung in die Prozeßdatenverarbeitung, Hanser-Verlag, 1996
- Wittgruber: Digitale Schnittstellen und BUS-Systeme, Vieweg-Verlag, 1999
- Tanenbaum, Andrew: Moderne Betriebssysteme, Pearson Studium, 3. Auflage, 2009
- Witzak, Michael: Echtzeit Betriebssysteme, Franzis Verlag, 2000
- Baumgarten, Bernd: Petri-Netze, Wissenschaftsverlag, 1990
- Labrosse, Jean: MicroC/OS-II, CMP Books, 2002
- Quade, Jürgen; Mächtel, Michael: Moderne Realzeitsysteme kompakt, dpunkt.verlag, 2012
- Labrosse, Jean: MicroC/OS-II, CMP Books, 2002

Studiengänge:

- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB368 – Venture Lab

Verantwortliche:	Jan-Paul Lüdtkke
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB389 – Venture Lab
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Portfolio-Prüfung
Prüfungsdauer:	15 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	45 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	105 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Valena Maiwald

Studieninhalte:

Das Modul gliedert sich in einen wissensvermittelnden und einen anwendungsorientierten Teil.

Ergänzend dazu können die Studierenden erweiterte Angebote in Anspruch nehmen, die fundiertes Wissen für eine angestrebte Gründung vermitteln.

Zu folgenden Themen wird Wissen vermittelt:

- Einführung in Entrepreneurship
- Entstehung innovativer Geschäftsideen
- Zielgruppendefinition und Persona-Methodik
- Value Proposition Design
- Validierung des Problem-Solution-Fits
- Rapid Prototyping
- Business Modell
- Markt und Wettbewerb
- Pitching

Im Rahmen des Moduls werden die Studierenden befähigt, eigenständig und in Begleitung durch Gründungsberatung zu arbeiten und die obengenannten Themen in Teams auf die eigene Gründungsidee zu transferieren und auszuarbeiten. Ergänzt wird dies durch sogenannte Toolboxes, die bei starker Gründungsmotivation besucht werden können. Themen sind dabei zum Beispiel:

- Kalkulation und Preisstrategie
- Startup-Finanzierung und Förderung
- Marketing und Vertrieb
- Rechtsformen und Steuer

Lernergebnisse:

Die Studierenden...

... werden für Entrepreneurship als Karriereweg und Karrierewerkzeug sensibilisiert.

... verstehen den Wert unternehmerischen Denkens in einer Welt, die von hoher Veränderungsgeschwindigkeit, Unsicherheit, Komplexität und Mehrdeutigkeit geprägt ist.

...kennen Werkzeuge zur Identifikation von unternehmerischen Herausforderungen, geschäftlichen Opportunitäten oder bestehenden Problemen bei Zielgruppen.

...können unternehmerische Methoden für die Geschäftsmodellentwicklung identifizierten und anwenden.

...können mit Hilfe von Instrumenten der Lean-Startup-Methode eigene Gründungsvorhaben entsprechend von Zielgruppenbedürfnissen entwickeln, erproben und vor Kunden, Investoren und Partnern gewinnend vorstellen.

... arbeiten interdisziplinär zusammen und erschaffen innovative unternehmerische Vorhaben.

Voraussetzungen und Empfehlungen:

Zusammenhangswissen betriebswirtschaftlicher Grundlagen.

Literatur:

- Freiling, Jörg, Harima, Jan (2019): Entrepreneurship: Gründung und Skalierung von Startups, Gabler Verlag
- Fueglistaller, Urs et al. (2019): Entrepreneurship: Modelle - Umsetzung - Perspektiven Mit Fallbeispielen aus Deutschland, Österreich und der Schweiz, Gabler Verlag
- Hölzle/Tiberius/Surrey (2020): Perspektiven des Entrepreneurships: Unternehmerische Konzepte zwischen Theorie und Praxis
- Kailer/Weiß (2009): Gründungsmanagement kompakt, von der Idee zum Businessplan, Linde Verlag Wien
- Maurya, Ash (2012): Running Lean: Iterate from Plan A to a Plan That Works. Sebastopol/CA 2012
- Osterwalder, Alexander; Pigneur, Yves (2013) Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Hoboken/NJ
- Ries, Eric (2011): The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. New York/NY

Studiengänge:

- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB108 – Großintegrierte Systeme

Verantwortliche:	Dennis Säring
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB194 – Großintegrierte Systeme
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	60 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	75 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Dennis Säring

Teilleistung:	TB202 – Workshop Mikroprozessor
Lernform:	Projektarbeit
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	10 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Jörg Völker

Studieninhalte:

- Einführung in die Struktur vom Mikroprozessoren
- Befehlsstruktur
 - Datenflüsse in der Hardware
 - Befehlstypen
 - Adressierungsarten
 - IRQ-Struktur
- Einführung in die verwendeten Programmier-Tools
- Praktische Übungsaufgaben
 - Beginnend mit einfachen Aufgaben
 - In Folge steigender Schwierigkeitsgrad
- Abschlussaufgabe
- Technologische Aspekte und Abhängigkeiten bei der Prozessorentwicklung
- Grundkonzepte und Basistechnologien der Schaltkreisfamilien
- Speicher-Technologien
- Prozessorumgebung, Kommunikationskonzepte und Chipsätze
- Betrachtung ausgewählter Monocore-Prozessorstrukturen
- Multicore - Architekturen

Lernergebnisse:

Die Studierenden ...

- besitzen ein grundlegendes Verständnis für die realen, komplexen Implementierungen großintegrierter Funktionsgruppen, wie CPUs, Chipsets und Speichern
- verfügen über elementare physikalische Grundlagen der Schaltungstechnologien und der praktischen Implementierung großintegrierter Funktionselemente
- kennen die Komplexität der Strukturen von einfachen Zellen bis zum kompletten Design aktueller Mainstream-Prozessoren
- haben die Fähigkeit zur Einschätzung alternativer Entwicklungstrends in den Bereichen Speichertechnologie, Chipsatz und Prozessorstruktur

Studierende erlangen die ...

- Fähigkeit zur Erfassung der Struktur und Funktionalität eines realen, industrietypischen Mikroprozessors.
- Fähigkeit zum Erfassen der Struktur und der Eigenschaften einer realen Assemblersprache.
- Kompetenzen zum Schreiben und Austesten von Assemblerprogrammen an realitätsnahen Problemstellungen.

Voraussetzungen und Empfehlungen:

Grundlegendes Verständnis von Digitaltechnik

Literatur:

- Tanenbaum: Computerarchitecture, 4. Auflage, Addison-Wesley
- Hennessy, Patterson: Computer architecture, Morgan Kaufman Publishing
- Diverse aktuelle weblinks für die jeweilige Veranstaltung
- SCHMITT, Günter: Mikrocomputertechnik mit den Prozessoren der 68000-Familie. Maschinenorientierte Programmierung. Grundlagen - Schaltungstechnik - Anwendungen. München: Oldenbourg, 1987

Studiengänge:

- Technische Informatik Bachelor of Science

◆ MB109 – Regelungstechnik

Verantwortliche:	Carsten Burmeister
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch/englisch

Bestandteile:

Teilleistung:	TB188 – Regelungstechnik
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	90 Min.
ECTS:	4.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	24 Stunden
Aufwand während Semesterferien:	54 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	Gesamtaufwand 108 Stunden weicht signifikant von erwarteten 4.0 ECTS ab.
Lehrende:	Carsten Burmeister

Teilleistung:	TB191 – Übg. Simulationssoftware
Lernform:	Übung
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	
ECTS:	1.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	1 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	7 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	22 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	29 Stunden
Lehrende:	Carsten Burmeister

Studieninhalte:

- Einführung in die Regelungstechnik
 - Anwendungsbeispiele
 - Regelziele Genauigkeit, Dynamik, Stabilität
- Grundlagen der Systemtheorie
 - Strukturdiagramme
 - Linearisierung und Normierung
 - Lösung linearer Differentialgleichungen durch Laplace-Transformation
- Systemfunktionen
 - Einfache Funktionen: P-, I-, D-, Tt-Glieder
 - Zusammengesetzte Funktionen: PT1-, DT1-, PT2-Glieder
 - Lineare Regelalgorithmen: P-, I-, PI-, PID-Regler
 - Unstetige Regler: Zweipunkt-, Dreipunktregler ohne und mit Rückkopplungen
- Analyse von Regelkreisen
 - Übertragungsfunktionen offener und geschlossener Kreise
 - Regelung einfacher Kreise
 - Beurteilung von Regelzielen
- Stabilität
 - Wurzelortungsverfahren
 - Das Nyquist-Kriterium
 - Frequenzkennlinienverfahren
- Empirische Verfahren
 - Ziegler-Nichols Verfahren
 - Wendetangentenverfahren nach Schwarze

-
- Einführung in MATLAB / Octave + Simulink
 - Entwicklung von Ortskurven zum Umgang mit komplexen Zahlen
 - Lösung linearer Systeme
 - numerische Lösung technischer Problemstellungen

Lernergebnisse:

Die Studierenden haben ...

- Kenntnis der grundlegenden Ziele der Regelungstechnik.
- die Fähigkeit zur Beurteilung von Stabilität, Dynamik und Genauigkeit offener und geschlossener Kreise.
- Kenntnis über Rechenverfahren im Zeit- und Bildbereich zur Dimensionierung von Regeleinrichtungen und zum Nachweis der Regelziele.
- die Fähigkeit, eindimensionale Systeme zu analysieren, passende Regelalgorithmen für vorgegebene Aufgabenstellungen zu entwerfen und sie bezüglich des Erreichens von Regelzielen zu beurteilen.

Die Studierenden haben ...

- die Fähigkeiten im Umgang mit Numerik- und Simulations-Software.
- Kenntnisse über die Arbeitsweise und Umfang industrietüblicher Numerik- und Simulationssoftware.
- Fähigkeiten der typischen Programmierung in den jeweiligen Script-Sprachen.
- die Fähigkeit mit den Methoden und Techniken zur Visualisierung numerischer Probleme umzugehen.

Voraussetzungen und Empfehlungen:

- Fundierte Kenntnisse in höherer Mathematik (Differentialgleichungen, lineare Algebra, komplexe Zahlen)
- Sicherer Umgang mit der Laplace-Transformation und ihren Anwendungen
- Grundkenntnisse in Systemtheorie und Signalverarbeitung
- Grundkenntnisse in Elektrotechnik oder Mechanik sind von Vorteil

Literatur:

- BEUCHER, Ottmar: MATLAB und Simulink: Grundlegende Einführung für Studenten und Ingenieure in der Praxis. 4. Aufl., Pearson Studium, 2008
- OKORO, Ogbonnaya Inya; CHIKUNI, Edward: The Essential MATLAB & Simulink: For Engineers and Scientists. Juta Legal and Academic Publishers, 2009
- Lunze, J.: Regelungstechnik 1 - Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen, Springer Verlag, 2010
- Zacher, S., Reuter, M.: Regelungstechnik für Ingenieure - Analyse, Simulation und Entwurf von Regelkreisen, Springer Verlag 2014

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB135 – Projekt Eingebettete Systeme

Verantwortliche:	Timm Bostelmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB196 – Prakt. PCB-Design
Lernform:	Praktikum
Prüfungsform:	Abnahme
Prüfungsdauer:	20 Min.
ECTS:	1.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	3 Termine
Zeit in Veranstaltungen:	11 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	19 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	30 Stunden
Lehrende:	Timm Bostelmann

Teilleistung:	TB198 – Prakt. Schaltungstechnik
Lernform:	Praktikum
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	
ECTS:	1.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	4 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	15 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	30 Stunden
Lehrende:	Thomas Starke

Teilleistung:	TB199 – Projekt Mikrocontroller
Lernform:	Projektarbeit
Prüfungsform:	Abnahme
Prüfungsdauer:	30 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	30 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Timm Bostelmann

Studieninhalte:

Projekt Mikrocontroller:

- Praktische Unterweisung an den Laborgeräten
- Planung der Projektdetails und des Projektablaufs
- Hardware-Schaltungsentwurf inklusive eines Mikrocontrollers
- Softwareerstellung für den eingesetzten Mikrocontroller
- Inbetriebnahme der Schaltung auf einem Steckbrett
- Test und empirischer Nachweis der Funktion
- Technische Dokumentation des Projektes

Prakt. PCB-Design:

- Leiterplattentechniken
- Einführung in ein PCB-Tool
- Durchführung eines Leiterplattenentwurfs
- Fertigung der Leiterplatte
- Inbetriebnahme der Leiterplatte

Prakt. Schaltungstechnik:

- Wechselnde, ausgewählte Versuche aus dem Stoffgebiet der Vorlesungen Elektronik und Halbleiterschaltungstechnik

Lernergebnisse:

Die Studierenden...

- können sich in ein vorgegebenes oder selbst gewähltes Projektthema einarbeiten und dessen Realisierung planen.
- können analoge und digitale Hardware (je nach Projektthema) als Schnittstelle eines Mikrocontrollers entwerfen.
- können Software für einen Mikrocontroller entwickeln, testen und in Betrieb nehmen.
- können ihre Lösung auf System- und Implementierungsebene in einer schriftlichen Ausarbeitung dokumentieren.
- kennen Entwurfsregeln für Leiterplatten.
- sind in der Lage sich in eine Leiterplattenentwurfssoftware einzuarbeiten.
- können eine Leiterplatte entwickeln und bestücken.
- kennen Regeln zum Umgang mit elektronischen Komponenten.
- kennen Vor- und Nachteile unterschiedlicher Messmethoden.
- können Messreihen durchführen, protokollieren und auswerten.

Voraussetzungen und Empfehlungen:

Kenntnisse in Elektronik, Halbleiterschaltungstechnik und systemnaher Programmierung mit C werden vorausgesetzt.

Literatur:

Datenblätter der verwendeten Komponenten (Projektabhängig)
Graham Keith, Jon Evans, Glenn Peterson: Getting Started in KiCad, The KiCad Team, 2023
Ulrich Tietze, Christoph Schenk: Halbleiterschaltungstechnik, 16. Auflage Springer Verlag, 2016

Studiengänge:

- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB148 – Seminar Technische Informatik

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB040 – Seminar
Lernform:	Seminar
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	30 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	150 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 165 Stunden weicht signifikant von erwarteten 5,0 ECTS ab.</i>
Lehrende:	Franziska Bönke Gunnar Harms Alexander Fischer Thorsten Giersch Stefan Christoph Weber

Studieninhalte:

Fachvorträge mit anschließender Gruppendiskussion.

Lernergebnisse:

Die Studierenden

- können wissenschaftliche Themen angemessen strukturieren
- können eine eigenständige Zielsetzung erarbeiten und umsetzen
- können Inhalte recherchieren und übersichtlich aufbereiten
- können formale Kriterien sicher beachten und anwenden
- können eine schriftliche Ausarbeitung größeren Umfangs erstellen
- können kontroverse Lehrmeinungen und aktuelle Trends zu einem Thema herausarbeiten
- können ihre Ergebnisse in angemessener Form vortragen und mit den Seminarteilnehmern diskutieren

Voraussetzungen und Empfehlungen:

Fachliche und persönliche Kompetenzen der zurückliegenden Semester, insbesondere themenabhängig fachverwandte Module und "Soft Skills"

Literatur:

Recherche nach aufgabenbezogener Literatur, teilweise aufgabenspezifische Vorgabe einzelner Literaturquellen.

Empfehlungen zur Einführung in das wissenschaftliche Arbeiten

- Axel Bänsch, Dorothea Alewell, Wissenschaftliches Arbeiten, 11. Aufl., München [u.a.]: Oldenbourg 2013.

- Werner Heister, Dagmar Weßler-Poßberg, Studieren mit Erfolg: Wissenschaftliches Arbeiten für Wirtschaftswissenschaftler, 2., überarbeitete Auflage, Stuttgart: Schäffer-Poeschel 2011.
- Jens Hiller, Arbeitstechniken und wissenschaftliches Arbeiten, Herne: Kiehl 2017.
- Walter Krämer, Wie schreibe ich eine Seminar- oder Examensarbeit. 3., überarbeitete und aktualisierte Aufl., Frankfurt: Campus 2009.
- Lydia Prexl, Mit digitalen Quellen arbeiten. Richtig zitieren aus Datenbanken, E-Books, YouTube & Co., 2., aktualisierte und erweiterte Aufl., Paderborn: Ferdinand Schöningh (UTB) 2016.
- Manuel René Theisen, Wissenschaftliches Arbeiten: Technik - Methodik Form, 15. Aufl., München: Vahlen 2011.

Studiengänge:

- Technische Informatik Bachelor of Science

◆ MB366 – Datenbanken

Verantwortliche:	Marco Pawlowski
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB361 – Datenbanken
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	60 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	30 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 45 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Marco Pawlowski

Teilleistung:	TB362 – Übg. Datenbanken
Lernform:	Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	20 Min.
ECTS:	2.0
Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester
Dauer (pro Termin):	1 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	7 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	52 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	59 Stunden
Lehrende:	Lars Neumann

Studieninhalte:

Vorlesungsbegleitende praktische Übungen in SQL und zum Datenbankentwurf

- Einführung in die Datenbanktechnologie
- Datenbanksprache SQL - Einführung
- Datenbank-Abfrage mit SQL
- Datenbanksprache SQL - Einrichten der Datenbank
- Das Entity-Relationship-Datenmodell
- Das Relationale Datenmodell
 - Relationenschemata und Datenabhängigkeiten
 - Relationale Datenbanken
 - Normalformen
- Datenbank - Lebenszyklus

Lernergebnisse:

Die Studierenden ...

- besitzen die Fähigkeit, ein Datenbanksystem mit SQL zu befragen und in nicht-triviale textuelle Anfrageanforderungen in SQL zu überführen.
- haben grundlegende Kenntnisse über die Ausführung der von ihnen gestellten Anfragen.
- haben die Kompetenz, ein Datenbankentwurfswerkzeug grundlegend zu bedienen.

Die Studierenden ...

- beherrschen die Grundlagen der relationalen Datenbanktechnologie;
- erlangen die Fähigkeit, selbstständig einen Datenbankentwurfsprozess zu planen, eine relationale Datenbank unter Nutzung von SQL einzurichten und die Informationsverarbeitung mittels relationaler Datenbanksysteme unter Nutzung von SQL durchzuführen;
- erlangen die Fähigkeit, mit einem Entwurfstool einen Datenbankentwurfsprozess durchzuführen und mittels SQL selbstständig Anfragen an ein Datenbanksystem zu stellen.

Voraussetzungen und Empfehlungen:

Vorausgesetzt wird ein grundlegendes Verständnis der Konzepte von Programmiersprachen.

Empfohlen wird die Einrichtung der in der Übung verwendeten Werkzeuge.

Literatur:

- Elmasri, Ramez; Navathe, Shamkant B.: Grundlagen von Datenbanksystemen. 3. Aufl. München: Pearson -Verlag, 2009.
- Meier, Andreas: Relationale Datenbanken Leitfaden für die Praxis. Berlin: Springer-Verlag, 2004.
- Vetter, Max: Aufbau betrieblicher Informationssysteme mittels konzeptioneller Datenmodellierung. 8. Aufl. Stuttgart: Vieweg-Teubner, 1998.
- Vossen, Gottfried: Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme. 5. Aufl. Oldenbourg: Oldenbourg-Wissenschaftsverlag, 2008.

Vorlesungsunterlagen

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB373 – Robotik

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB086 – Prakt. Robotik
Lernform:	Praktikum
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	10 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	15 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 105 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Hermann Höhne

Teilleistung:	TB368 – Robotik
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	60 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	15 Stunden
Aufwand während Semesterferien:	15 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Ulrich Hoffmann

Studieninhalte:

Inhalte der Vorlesung:

- Strukturen der Fertigungstechnik
- Flexible Fertigungszellen
- Industrieroboter
- Strukturen und Aufbau von Robotern
- Kinematik
- Antriebe
- Effektoren
- Steuerstrategien
- Koordinatentransformationen
- Punkt-zu-Punkt-Steuerung
- Steuerung mit Interpolation
- Mensch-Maschine-Kommunikation
- Roboter-Programmiersysteme
- Roboter-Sprachen
- Intelligente Sensorik
- Integration von Optischen Sensoren

Anhand eines Projekts werden die Inhalte aus der Vorlesung praktisch umgesetzt. Schwerpunkt liegt bei den Grundlagen zur Programmierung mobiler Roboter bis hin zu einfachem autonomen Fahren.

Lernergebnisse:

Die Studierenden ...

- besitzen fundierte Kenntnisse der technischen Grundlagen der Robotik.
- besitzen ausgehend von den Entwicklungstendenzen im Bereich der Flexiblen Fertigungstechnik die grundlegende Kompetenz für das Verständnis für Funktionsweisen und Einsatzschwerpunkte von Industrierobotern.
- verfügen über die Kompetenz, die Möglichkeiten der Verbindung von Robotern mit "intelligenten" Sensoren zu durchdringen, insbesondere die Erkennung und Einschätzung der Eigenschaften optischer Sensorsysteme.
- können verschiedene Konzepte der Offline-Programmierung von Industrierobotern identifizieren.

- verstehen aktuelle Entwicklungstendenzen zur Erhöhung der Selbständigkeit bei Robotern.

Nach Bearbeitung des Praktikums sind die Studierenden in der Lage ...

- Lerninhalte der Vorlesung im Rahmen eigener Erfahrungen zu vertiefen.
- Sich selbstständig in eine neue Programmierungsumgebung einzuarbeiten.
- Ein gegebenes Roboterprogrammiersystem zu nutzen.
- Techniken zum Sammeln, Glätten und Bewerten von Sensordaten anzuwenden.
- Typische Problemstellungen mittels einfacher Algorithmen zu lösen.
- Eigene Ideen und Lösungsansätze zu implementieren.
- Versuchsergebnisse in einer schriftlichen Dokumentation zu präsentieren.

Voraussetzungen und Empfehlungen:

- Kenntnisse in Linearer Algebra und Trigonometrie
- Grundlegende Fähigkeiten imperative Programme zu erstellen und auf Software-Bibliotheken zuzugreifen
- Grundkenntnisse und grundlegende Fähigkeiten in der Programmierung von Bildverarbeitungsalgorithmen und der Benutzung einschlägiger Bibliotheken.

Literatur:

- McKerrow: Introduction to Robotics, Addison Wesley, 1990
- Wirth: Flexible Fertigungssysteme, Hüthig-Verlag
- Vukobratovic: Introduction to Robotics, Springer, 1995
- Blume, Dillmann: Frei Programmierbare Roboter, Vogel Verlag
- Blume, Jakob: Programmiersprachen für Industrieroboter, Vogel Verlag, 1985
- Hertzberg, J: "Mobile Roboter: Eine Einführung aus Sicht der Informatik", eXamen.press, 2012
- Prat: "Sensordatenfusion und Bildverarbeitung zur Objekt- und Gefahrenerkennung", 2010
- Quigley: "Programming Robots with ROS", 2016

Studiengänge:

- IT-Ingenieurwesen Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science

◆ MB118 – Soft Skills

Verantwortliche:	Jan-Paul Lüdtkke
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB042 – Assistenz
Lernform:	Assistenz
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	
ECTS:	3,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	3 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	22 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	68 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Jan-Paul Lüdtkke

Teilleistung:	TB043 – Communication Skills
Lernform:	Workshop
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	
ECTS:	2,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Anna-Magdalena Kölzer

Studieninhalte:

Communication Skills

Im Rahmen des Workshops werden die folgenden Inhalte behandelt:

- Selbstanalyse (Fragebogen) als Grundlage für Bewerbungen
- Rhetorik & Präsentation (Theorie und Praxis)
- Struktur und Aufbau von Bewerbungsunterlagen
- Bewerbungsprozess
- Interview (Theorie und Praxis)
- Assessment Center (Theorie)
- Persönlichkeitsfragebogen und Testverfahren (Intelligenz und Konzentration) (Praxis)
- Gruppenübungen (Praxis)

Assistenz

Im Rahmen der Assistenz werden die Studierenden von den Hochschullehrenden mit konkreten (Teil-)Projekten betraut. Diese können ein weites Spektrum umfassen. So sind z.B. die Durchführung kleinerer empirischer Umfragen oder auch die eigenständige Recherche und Ausarbeitung spezieller Fachinhalte denkbar. Ebenso in Betracht kommen die Durchführung von Tutorien oder Übungen. Die Assistenz ist selbstständig zu bearbeiten und kann die Abstimmung mit anderen Studierenden erfordern.

Lernergebnisse:

Nach Abschluss des Moduls haben die Studierenden die Fähigkeit erworben, in Kooperation mit den Dozenten und Assistenten, ihr Wissen und ihre Erfahrungen aus früheren Veranstaltungen der Betriebswirtschaftslehre, Mathematik und Informatik an Studierende jüngerer Semester weiter zu geben. Mit zunehmender Dauer des Semesters verbinden die Studierenden Kenntnisse aus dem Workshop "Communication Skills" mit ihrer Assistenz Tätigkeit.

Die Studierenden verfügen nach dem Besuch des Workshops über folgende Kompetenzen:

- Besitz verbesserter persönlicher Soft Skills, wie sie für Studium oder Beruf erforderlich sind
- Sensibilität für menschliche Interaktionen und Betriebsprozesse
- Besitz erweiterter rhetorischer Fähigkeiten im Rahmen von Präsentationen, Vorträgen und Referaten sowie sozialer Kompetenz

- Kenntnis der Bedeutung von verbalen und nonverbalen Signalen für die eigene Kommunikation sowie die Fähigkeit, diese zu erkennen
- Fähigkeit zum angemessenen Verhalten bei Teamarbeit oder Projekten
- Fähigkeit zur Selbstdarstellung bei Bewerbungen, Interviews, Assessment-Centern.

Die Studierenden entwickeln im Rahmen der Assistenz unter Anleitung eines Hochschullehrers die Fähigkeiten ...

- fachspezifische Aufgabenstellungen zu analysieren
- problemspezifische Lösungen zu konzipieren und
- als Ergebnis begründet zu präsentieren.

Voraussetzungen und Empfehlungen:

Fachliche Inhalte der ersten vier Studiensemester

Literatur:

- ARNOLD, Frank:
Management von den besten lernen.
München: Hans Hauser Verlag, 2010
- APPELMANN, Björn:
Führen mit emotionaler Intelligenz.
Bielefeld: Bertelsmann Verlag, 2009
- BIERKENBIEHL, Vera F.:
Rhetorik, Redetraining für jeden Anlass. Besser reden, verhandeln, diskutieren.
12. Aufl. München: Ariston Verlag, 2010
- BOLLES, Nelson:
Durchstarten zum Traumjob. Das ultimative Handbuch für Ein-, Um- und Aufsteiger.
2. Aufl. Frankfurt/New York: Campus Verlag, 2009
- DUDENREDAKTION mit HUTH, Siegfried A.:
Reden halten - leicht gemacht. Ein Ratgeber.
Mannheim/Leipzig: Dudenverlag, 2007
- GRÜNING; Carolin; MIELKE; Gregor:
Präsentieren und Überzeugen. Das Kienbaum Trainingskonzept.
Freiburg: Haufe-Lexware Verlag, 2004
- HERTEL, Anita von:
Professionelle Konfliktlösung. Führen mit Mediationskompetenz.
Handelsblatt, Bd., 6, Kompetent managen.
Frankfurt: Campus Verlag, 2009
- HESSE, Jürgen; SCHRADER, Hans Christian:
Assessment-Center für Hochschulabsolventen.
5. Auflage, Eichborn: Eichborn Verlag, 2009
- MENTZEL, Wolfgang; GROTZFELD, Svenja; HAUB, Christine:
Mitarbeitergespräche.
Freiburg: Haufe-Lexware Verlag, 2009
- MORITZ, Andr; RIMBACH, Felix:
Soft Skills für Young Professional. Alles was Sie für ihre Karriere wissen müssen.
2. Aufl. Offenbach: Gabal Verlag, 2008
- PERTL, Klaus N.:
Karrierefaktor Selbstmanagement. So erreichen Sie ihre Ziele.
Freiburg: Haufe-Verlag, 2005
- PORTNER, Jutta:
Besser verhandeln. Das Trainingsbuch.
Offenbach: Gabal Verlag, 2010

- PÜTTJER, Christian; SCHNIERDA, Uwe:
Assessment-Center. Training für Führungskräfte.
Frankfurt/New York: Campus Verlag, 2009
- PÜTTJER, Christian; SCHNIERDA, Uwe:
Das große Bewerbungshandbuch.
Frankfurt: Campus Verlag, 2010
- SCHULZ VON THUN, Friedemann; RUPPEL, Johannes; STRATMANN, Roswitha:
Miteinander Reden. Kommunikationspsychologie für Führungskräfte.
10. Auflage, Reinbek bei Hamburg: rororo, 2003

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB122 – IT-Sicherheit

Verantwortliche:	Gerd Beuster
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch/englisch

Bestandteile:

Teilleistung:	TB048 – IT-Sicherheit
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	90 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Gerd Beuster

Studieninhalte:

- Gegenstandsbereich der IT-Sicherheit
- Aktuelle Richtlinien, Standards, Normen und Gesetze
- Bedrohungen der IT-Sicherheit und daraus resultierende Risiken
- Primäre Sicherheitsziele
- Überblick über Verfahren zur Erreichung der Ziele
- Kryptografische Verfahren
 - Verschlüsselungsverfahren
 - Symmetrische Verschlüsselungsverfahren
 - Asymmetrische Verschlüsselungsverfahren
 - Hash-Funktionen
 - Schlüsselmanagement
 - Zertifikate
 - Kryptografische Protokolle
 - Digitale Signatur
 - Zeitstempel
 - TLS-Protokoll
- Authentifizierungsverfahren
- Übertragungssicherheit in Netzen
 - Sichere IP-Kommunikation
 - VPN-Technologien
- Sicherheitsarchitekturen und ihre Komponenten
 - Sicherheitsaspekte von Web-Servern
 - Firewall-Systeme
 - Intrusion Detection-Systeme
- Sicherheit von Web-Anwendungen
- Sicherheit bei Nutzung von Large Language Models (LLMs)
- Technisch / organisatorische Maßnahmen zur Erhöhung der IT-Sicherheit
- Risiko- und Sicherheitsmanagement

Lernergebnisse:

Die Studierenden erwerben die notwendigen Kenntnisse, um Softwaresysteme und ihre betrieblichen Einsatzszenarien in Hinblick auf ihre Sicherheit einschätzen zu können. Sie sind in der Lage, bei der

Konzeption und Entwicklung von Softwaresystemen und in ihrem Unternehmenseinsatz relevante Sicherheitsaspekte zu berücksichtigen.

- Kenntnis der unterschiedlichen Bedrohungsszenarien und -arten.
- Kenntnis der besonderen Gefahren bei internetbasierten Anwendungen.
- Kenntnis typischer primärer Sicherheitsziele (Vertraulichkeit, Authentifizierung, Verbindlichkeit, u.a.).
- Kenntnis der Verfahren zur Gewährleistung der unterschiedlichen Sicherheitsziele.
- Kenntnis der praxisrelevanten kryptografischen Verfahren und Protokolle.
- Kenntnis der Sicherungsmaßnahmen in Rechnernetzen.
- Fähigkeit, grundlegende Sicherungsmaßnahmen für Web-Anwendungen umzusetzen.
- Fähigkeit, Large Language Models (LLMs) sicher in Software-Systeme einzubinden
- Fähigkeit, die Qualität von durch LLMs generiertem Code einzuschätzen
- Kenntnis der Bestandteile einer IT-Sicherheitsinfrastruktur und ihrer zentralen Funktionalitäten.
- Kenntnis der Verfahren zur Risikoabschätzung und Bewertung der Sicherheit von IT-Systemen und die Fähigkeit, diese anzuwenden.

Voraussetzungen und Empfehlungen:

Das Modul setzt grundlegende Kenntnisse der Programmierung und des Aufbaus eines Computersystems sowie von Computernetzen voraus.

Literatur:

- Anderson, Ross J.: Security Engineering : A Guide to Building Dependable Distributed Systems. 3. Auflage. Hoboken (NJ), USA: Wiley & Sons, 2020.
- BSI - Bundesamt für Sicherheit in der Informationstechnik (Hrsg.): Informationssicherheit und IT-Grundschutz : BSI-Standards 200-1, 200-2 und 200-3. 1. Auflage. Bonn : BSI, 2017.
- Eckert, Claudia: IT-Sicherheit : Konzepte - Verfahren - Protokolle. 11. Auflage. München : Oldenbourg, 2023.
- Ferguson, Niels; Schneier Bruce, Kohno; Tadayoshi: Cryptography Engineering : Design Principles and Practical Applications. Hoboken (NJ), USA: Wiley & Sons, 2010.
- Kersten, Heinrich; Klett, Gerhard: Der IT Security Manager. 4. Auflage. Wiesbaden: Springer Vieweg, 2015.
- Pfleeger, Charls P.;Pfleeger, Shari Lawrence: Security in Computing. 6. Auflage. München: Prentice Hall, 2024.
- Progutke, Werner: Basiswissen IT-Sicherheit : Das Wichtigste für den Schutz von Systemen & Daten. 3. Auflage. Heidelberg: Springer Campus, 2017.
- Stallings, William: Computer Security : Principles and Practice. 5. Auflage. München: Pearson, 2024.
- Stallings, William: Cryptography and Network Security : Principles and Practice. 8. Auflage. München: Pearson, 2023.
- Swoboda, Joachim; Spitz, Stephan; Pramateftakis, Michael: Kryptographie und IT-Sicherheit : Grundlagen und Anwendungen. 2. Auflage Wiesbaden: Vieweg + Teubner Verlag, 2011.

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Medieninformatik Bachelor of Science

- Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
-

◆ MB133 – Laborprojekt

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB046 – Projektmanagement
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	60 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Gerrit Remané

Teilleistung:	TB195 – Laborprojekt
Lernform:	Projektarbeit
Prüfungsform:	Abnahme
Prüfungsdauer:	60 Min.
ECTS:	8.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	20 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel einteilbarer Aufwand:	170 Stunden
Gesamtaufwand:	240 Stunden
Lehrende:	Sergei Sawitzki

Studieninhalte:

- Einführung in Projektmanagement
- Projektphasen (Initiierung, Planung, Durchführung, Abschluss)
- Soft Skills (Motivation, Veränderungsmanagement, Feedback, ...)
- Spezifische Ansätze (Wasserfall, Agil, Großprojekte, ...)

Der Inhalt ist themenabhängig. Bisherige Projektthemen sind zum Beispiel:

- HV-Programmier- und Testgerät für AVR Mikrocontroller
- Funktionserweiterungen für das FPGA-Entwicklungssystem (Audiiodaten-Verarbeitung, analoge Ein- und Ausgabe)
- Steuerungsplatine für bürstenlose Motoren mit Drehzahlregelung
- Erprobung des Einsatzes von Evolutionsstrategien für die Prozessoptimierung im Prozessrechnerlabor

Lernergebnisse:

Die Studierenden

- können in einem fortgeschrittenen Semester die bis dahin erworbenen Kenntnisse und Fertigkeiten einsetzen
- können ein Projekt von der ersten Spezifikation bis zum funktionsfähigen Prototypen inklusive Erstellung der technischen Dokumentation durchzuführen
- verstehen die spezifischen Charakteristika und Herausforderungen von Projekten (z.B. im Unterschied zu Prozessen)
- können die wichtigsten Projektmanagement-Tools je Projektphase anwenden (Initiierung, Planung, Durchführung, Abschluss)
- können wesentliche Konzepte und Methoden anwenden, um Mensch-bezogene Herausforderungen im Projektumfeld zu analysieren und zu lösen (z.B. Motivation, Feedback, Veränderung)
- können geeignete Projektmanagement-Ansätze (Wasserfall vs. Agil) je nach Projekttyp auswählen

Voraussetzungen und Empfehlungen:

Kenntnisse und Fertigkeiten aus zurückliegenden Projekten und Übungen, abhängig von der Aufgabenstellung

Literatur:

abhängig von der Aufgabenstellung

- Verzuh: The Fast Forward MBA in Project Management, Fifth Edition, New Jersey, 2016
- Wysocki: Effective Project Management – Traditional, Agile, Extreme, Seventh Edition, Indianapolis, 2014
- PMI: A Guide to the Project Management Body of Knowledge (Pmbok Guide), Sixth Edition, Newton Square, 2017

Studiengänge:

- Technische Informatik Bachelor of Science

◆ MB134 – Diskrete Systeme

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB193 – Diskrete Regelungstechnik
Lernform:	Vorlesung
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	60 Min.
ECTS:	1,5
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	10 Stunden
Aufwand während Semesterferien:	5 Stunden
Flexibel einteilbarer Aufwand:	15 Stunden
Gesamtaufwand:	45 Stunden
Lehrende:	Carsten Burmeister

Teilleistung:	TB197 – Prakt. Rechnergestützter Entwurf digitaler Systeme, Rechnergestützter Entwurf digitaler Systeme
Lernform:	Praktikum
Prüfungsform:	Mündliche Prüfung
Prüfungsdauer:	60 Min.
ECTS:	3,5
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	15 Stunden
Aufwand während Semesterferien:	15 Stunden
Flexibel einteilbarer Aufwand:	60 Stunden
Gesamtaufwand:	105 Stunden
Lehrende:	Sergei Sawitzki

Studieninhalte:

- Digitale Regelung
- Abtastsysteme
- Diskrete Regelalgorithmen (Direct Digital Control)
- Fuzzy-Regelung
- Entwurfsprozess, Entwurfsfluss und Entwurfsschritte
 - Systematik und Einordnung
 - X- und Y-Diagramm
 - Schaltungstechnologien und Entwurfsstile
- Synthese und Logikoptimierung
- Technology Mapping
- Partitionierung
- Floorplanning
- Platzierung
- Verdrahtung
 - Globalverdrahtung
 - Detailverdrahtung
- Modellierung und Analyse der Zeitverhaltens
- Fehlermodellierung und Testsatzerzeugung
- Vorstellung der Aufgabenstellung
- Einarbeitung in die Entwurfswerkzeuge
- Umsetzung und Dokumentation der Aufgabe

Lernergebnisse:

Die Studierenden vertiefen die in der Lehrveranstaltung Rechnergestützter Entwurf digitaler Systeme erworbenen Kompetenzen durch Implementierung eines Entwurfsschritts in Software und quantitative Analyse der Ergebnisse.

Die Studierenden ...

- haben ein tiefgründiges Verständnis der Methoden und Algorithmen des modernen Schaltungs- und Systementwurfs und der Testung
- kennen die in den modernen Entwurfssystemen eingesetzten EDA-Algorithmen
- begreifen den Schaltungsentwurfsprozess von der Spezifikation bis zur technischen Umsetzung und können in diesen bei Bedarf gezielt eingreifen
- können neue EDA-Algorithmen entwerfen bzw. bestehende verändern
- können diskrete Methoden der Regelungstechnik zum energiesparenden Einsatz in Leistungsregelungen und zum Einsatz in der Prozessautomatisierung
- kennen die abstrakten Beschreibungsformen komplexer diskreter Regelsysteme
- können komplexe Systeme analysieren, in mehrdimensionalen Strukturen beschreiben und passende Regelalgorithmen für vorgegebene Aufgabenstellungen entwickeln

Voraussetzungen und Empfehlungen:

- Grundlegende Kenntnisse des Entwurfsflusses bei digitalem Systementwurf
- Programmierkenntnisse
- Grundlegende Kenntnisse der Regelungstechnik

Literatur:

- Sawitzki, Sergei: Einleitung und Aufgabenstellung zum Praktikum Rechnergestützter Entwurf digitaler Systeme, internes Dokument
- Lienig, Jens: Layoutsynthese elektronischer Schaltungen - Grundlegende Algorithmen für die Entwurfsautomatisierung, Springer Verlag 2006
- Sherwani, Naveed A.: Algorithms for VLSI Physical Design Automation, 3rd edition, Springer 1998
- Jansen, Dirk: Handbuch der Electronic Design Automation, Hanser Verlag 2001
- Alpert Charles J.; Mehta, Dinesh P.; Sapatnekar, Sachin S.: Handbook of Algorithms for Physical Design Automation, CRC Press 2009
- Hachtel Gary D.; Somenzi, Fabio: Logic Synthesis and Verification Algorithms, Springer 2006
- Betz, Vaughn; Rose, Jonathan; Marquardt, Alexander: Architecture and CAD for Deep-Submicron FPGAs, Kluwer Academic Publishers, 1999
- Franklin, G., Powell, J.: Feedback Control of Dynamic Systems. Pearson Education Limited 2019, 8. Auflage.

Studiengänge:

- Technische Informatik Bachelor of Science

◆ MB143 – Systementwurf mit VHDL

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB200 – Systementwurf mit VHDL
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	2.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	30 Stunden
Aufwand während Semesterferien:	10 Stunden
Flexibel einteilbarer Aufwand:	5 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Sergei Sawitzki

Teilleistung:	TB204 – Workshop VHDL
Lernform:	Workshop
Prüfungsform:	Abnahme
Prüfungsdauer:	30 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	75 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Timm Bostelmann

Studieninhalte:

- Systementwurf mit Hardwarebeschreibungssprachen, moderner Entwurfsfluss
- Modellaufbau
- Basiselemente
- Sprachkonstrukte
- Objekte
- Entwurfseinheiten
- Modellierungstechniken
- Simulation
- Weiterführende Konzepte (Dateien, Zeiger, Funktionen, Überladung)
- Synthese
- Einführungsaufgabe
- Lösung von Teilaufgaben, die am Ende zu einem Gesamtentwurf zusammengesetzt werden
- Test des entworfenen Systems auf einem FPGA-Board

Lernergebnisse:

Die Studierenden ...

- kennen die Rolle von Hardwarebeschreibungssprachen (HDL) im modernen Schaltungs- und Systementwurf
- verstehen den vollständigen auf HDL basierenden Entwurfsfluss
- können einfache bis mittelkomplexe digitale Systeme mit Hilfe der Hardwarebeschreibungssprache VHDL entwerfen
- kennen die Hardwarebeschreibungssprache VHDL mit ihren unterschiedlichen Entwurfsmethoden (funktionale Beschreibung, Registertransferbeschreibung, Strukturbeschreibung) und deren Möglichkeiten und Einschränkungen bezüglich der Schaltungssynthese
- können Simulation digitaler Systeme auf Basis von VHDL durchführen
- sind in der Lage, ein durch ein Pflichtenheft spezifiziertes Problem in eine reale Hardware umzusetzen

Voraussetzungen und Empfehlungen:

Kenntnisse der grundlegenden Bestandteile digitaler Systeme

Literatur:

- Reichardt, Jürgen; Schwarz, Bernd:
VHDL-Synthese: Entwurf digitaler Schaltungen und Systeme, 6. Auflage
Oldenbourg Verlag 2012
- Ashenden, Peter:
The Designers Guide to VHDL. 3rd edition
Elsevier 2008
- Lehmann, Gunther; Wunder, Bernhard; Selz, Manfred:
Schaltungsdesign mit VHDL
Franzis Verlag 1994

Studiengänge:

- Technische Informatik Bachelor of Science

◆ MB257 – Auslandssemester

Verantwortliche:	Herlinde Breckwoldt
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB039 – Auslandssemester
Lernform:	Ausland
Prüfungsform:	Ausland
Prüfungsdauer:	
ECTS:	30,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	25 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	187 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	900 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 1087 Stunden weicht signifikant von erwarteten 30,0 ECTS ab.</i>
Lehrende:	Herlinde Breckwoldt

Studieninhalte:

Für ein freiwilliges Auslandssemester ist der Umfang der zu leistenden ECTS-Punkte (bzw. der gleichwertige Umfang in lokalen Credits) in der jeweiligen Studienordnung vorgegeben. An der ausländischen Hochschule sind fachspezifische Kurse zu belegen, die mit dem in Wedel belegten Studiengang in ergänzendem Zusammenhang stehen. Das Studienprogramm wird vor der Abreise individuell mit dem International Office vereinbart.

Lernergebnisse:

Nach Abschluss des Auslandsemester besitzen die Studierenden ...

- fundierte Sprachkompetenzen in englischer, französischer oder spanischer Sprache.
- erweiterte Kenntnisse über die Kultur des Gastlandes.

Voraussetzungen und Empfehlungen:

Es wird empfohlen, mindestens eine der Sprachen zu beherrschen, die an der ausländischen Hochschule gesprochen wird.

Literatur:

Abhängig von der ausländischen Hochschule

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science

- IT-Management & Consulting Bachelor of Science
 - IT-Management / -Consulting & -Auditing Bachelor of Science
 - Marketing, Media & AI Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
 - Wirtschaftsingenieurwesen Bachelor of Science
-

◆ MB150 – Bachelor-Thesis

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	BTH – Bachelor-Thesis
Lernform:	Thesis
Prüfungsform:	Abschlussarbeit
Prüfungsdauer:	30 Min.
ECTS:	12,0
Benotung:	Zehntelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	0 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	0 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	200 Stunden
Aufwand während Semesterferien:	140 Stunden
Flexibel einteilbarer Aufwand:	20 Stunden
Gesamtaufwand:	360 Stunden
Lehrende:	Sergei Sawitzki

Studieninhalte:

Die Bachelor-Thesis soll im Regelfall in Kooperation mit einem Unternehmen erarbeitet werden. Themen aus den Arbeitsgruppen und Laboren der Hochschule sind ebenfalls möglich. Die Arbeit ist als abschließende, vom Studierenden eigenständig aber hochschul- und unternehmensseitig betreutes Projekt zu verstehen. Im Sinne der Zielsetzung der Bachelor-Ausbildung, der Erlangung des ersten berufsqualifizierenden Abschlusses, ist die Arbeit thematisch an einer Problemstellung eines kooperierenden Unternehmens orientiert oder sie besteht aus einer praxisrelevanten hochschulinternen Aufgabe.

Lernergebnisse:

Die Studierenden ...

- besitzen die Fähigkeit zur Durchführung einer praxisorientierten Arbeit
- können eine Fragestellung selbständig erarbeiten
- können die zu erarbeitende Problematik klar strukturieren
- können die Vorgehensweise und Ergebnisse in einer Ausarbeitung übersichtlich darstellen
- stärken ihre praktischen Fähigkeiten im Projektmanagement-Bereich und zur Selbstorganisation

Voraussetzungen und Empfehlungen:

Fachliche und persönliche Kompetenzen der zurückliegenden Semester, insbesondere themenabhängig fachverwandte Module und Seminar

Literatur:

themenabhängig

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science

- Computer Games Technology Bachelor of Science
 - Data Science & Artificial Intelligence Bachelor of Science
 - E-Commerce Bachelor of Science
 - Informatik Bachelor of Science
 - IT-Ingenieurwesen Bachelor of Science
 - IT-Management & Consulting Bachelor of Science
 - IT-Management / -Consulting & -Auditing Bachelor of Science
 - Marketing, Media & AI Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
 - Wirtschaftsingenieurwesen Bachelor of Science
-

◆ MB370 – Bachelor-Kolloquium

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB365 – Bachelor-Kolloquium
Lernform:	Kolloquium
Prüfungsform:	Kolloquium
Prüfungsdauer:	30 Min.
ECTS:	3.0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	1 Semesterwochenstunden
Termine im Semester:	1 Termine
Zeit in Veranstaltungen:	0 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	10 Stunden
Aufwand während Semesterferien:	10 Stunden
Flexibel einteilbarer Aufwand:	3 Stunden
Gesamtaufwand:	<i>Gesamtaufwand 23 Stunden weicht signifikant von erwarteten 3.0 ECTS ab.</i>
Lehrende:	Sergei Sawitzki

Studieninhalte:

- nach Thema der Bachelor-Arbeit unterschiedlich
- Fachvortrag über das Ergebnis der Bachelor-Arbeit
- Diskussion der Qualität der gewählten Lösung
- Fragen und Diskussion zum Thema der Bachelor-Arbeit und verwandten Gebieten

Lernergebnisse:

Die Studierenden ...

- besitzen die Fähigkeit der konzentrierten Darstellung eines intensiv bearbeiteten Fachthemas.
- verfestigen die Kompetenz, eine fachliche Diskussion über eine Problemlösung und deren Qualität zu führen.
- verfügen über ausgeprägte Kommunikations- und Präsentationsfähigkeiten.

Voraussetzungen und Empfehlungen:

Fachliche und persönliche Kompetenzen der zurückliegenden Semester, insbesondere themenabhängig fachverwandte Module und Bachelor-Thesis

Literatur:

themenabhängig

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science

- IT-Ingenieurwesen Bachelor of Science
 - IT-Management & Consulting Bachelor of Science
 - Marketing, Media & AI Bachelor of Science
 - Medieninformatik Bachelor of Science
 - Smart Technology Bachelor of Science
 - Technische Informatik Bachelor of Science
 - Wirtschaftsinformatik Bachelor of Science
 - Wirtschaftsingenieurwesen Bachelor of Science
-

◆ MB371 – Praktikum

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB366 – Praktikum
Lernform:	Praktikum
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	20 Min.
ECTS:	15,0
Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester
Dauer (pro Termin):	0 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	0 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	220 Stunden
Aufwand während Semesterferien:	220 Stunden
Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	450 Stunden
Lehrende:	Sergei Sawitzki

Studieninhalte:

- Sammeln von beruflichen Erfahrungen in einem der durch die Prüfungsverfahrensordnung vorgesehenen Tracks:
 - Business-Track, berufliche Tätigkeit in einem etablierten Unternehmen
 - Start-up-Track, Vorbereitung der Gründung eines eigenen Unternehmens
 - Project-Track, Teilnahme an einem größeren Projekt mit wechselnden Projektteams
 - Science-Track, detaillierte und forschungsorientierte Auseinandersetzung mit einem wissenschaftlichen Themenkomplexe
- Erstellung eines Praktikumsberichts
- Das berufsbildende Praktikum ist unabhängig vom Track im Umfang von 12 Wochen zu absolvieren

Lernergebnisse:

Die Studierenden

- erweitern ihre sozialen Kompetenzen und ihre Kontakte zu Unternehmen. Beides können sie nach ihrem Studiumsabschluss gewinnbringend für eine Bewerbung oder das Einleben bei ihrem späteren Arbeitgeber bzw. Gründung eines eigenen Unternehmens verwenden
- können Fach- und Methodenkompetenz auf ausgewählte Abläufe und Problemstellungen des betrieblichen Alltags zu übertragen

Voraussetzungen und Empfehlungen:

Fachliche und persönliche Kompetenzen der zurückliegenden Semester, insbesondere themenabhängig fachverwandte Module und "Soft Skills"

Literatur:

themenabhängig

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Betriebswirtschaftslehre Bachelor of Science
- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science