

Modulhandbuch Informatik Bachelor of Science

Version B_Inf25.1_W

Letzte Änderung: 2026-07-11 15:25:33

Inhaltsverzeichnis

MB001 – Analysis
MB002 – Mathematische Konzepte und Diskrete Mathematik
MB359 – Study Bootcamp Data Science
MB369 – Digitaltechnik
MB327 – Digitale Schaltungen und Prozessoren
MB344 – Objektorientierte Programmstrukturen
MB358 – Study Bootcamp Informatik
MB377 – Deskriptive Statistik und Lineare Algebra
MB040 – Algorithmen und Datenstrukturen
MB043 – Systemnahe Programmierung
MB116 – Technologie der Mediengestaltung und GUI-Programmierung
MB366 – Datenbanken
MB372 – Künstliche Intelligenz
MB378 – Fortgeschrittene Lineare Algebra
MB037 – Rechnernetze
MB057 – Fortgeschrittene Objektorientierte Programmierung
MB059 – Web-Anwendungen
MB097 – Bildbearbeitung und -analyse
MB122 – IT-Sicherheit
MB223 – Machine Learning
MB232 – Formale Sprachen
MB253 – Geschäftsprozesse mit ERP-Systemen
MB305 – Datenbanktheorie und -implementierung
MB036 – Programmierpraktikum
MB041 – Induktive Statistik
MB087 – Systemmodellierung
MB093 – Softwarequalität
MB098 – Anwendungsentwicklung in ERP-Systemen
MB118 – Soft Skills
MB123 – Prozessmodellimplementation
MB244 – Exploratory Data Analysis
MB268 – Projekt App- und Web-Development
MB293 – Digital Product Management
MB313 – Computer Vision
MB379 – Konzepte der funktionalen Programmierung
MB053 – Datenschutz und Medienrecht
MB058 – Software-Design
MB101 – Echtzeitsysteme
MB121 – Software-Projekt
MB257 – Auslandssemester
MB326 – Seminar Informatik
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:	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:	90 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	Gesamtaufwand 120 Stunden weicht signifikant von erwarteten 3.0 ECTS ab
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:	Gesamtaufwand 15 Stunden weicht signifikant von erwarteten 2.0 ECTS ab
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:	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:	60 Stunden
Aufwand während Semesterferien:	30 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	Gesamtaufwand 120 Stunden weicht signifikant von erwarteten 3,0 ECTS ab.
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.

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

◆ MB359 – Study Bootcamp Data Science

Verantwortliche:	Hendrik Annuth
Moduldauer:	2 Monate
Unterrichtssprache:	None

Bestandteile:

Teilleistung:	TB353 – Study Bootcamp Data Science
Lernform:	Projektarbeit
Prüfungsform:	Abnahme
Prüfungsdauer:	
ECTS:	15.0
Benotung:	Drittelnoten
Turnus:	Wintersemester
Dauer (pro Termin):	56 Semesterwochenstunden
Termine im Semester:	7 Termine
Zeit in Veranstaltungen:	245 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	100 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	70 Stunden
Gesamtaufwand:	415 Stunden
Lehrende:	Hendrik Annuth Emre Kilic Franziska Bönke Marco Pawlowski

Studieninhalte:

Das Modul *Study Bootcamp Data Science* ist als projektorientiertes Bootcamp konzipiert und vermittelt grundlegende Kompetenzen in Data Literacy, Data Analytics und Data Science. Im Mittelpunkt steht die Bearbeitung einer praxisnahen Problemstellung in Teamarbeit.

Studierende lernen, Geschäftsanforderungen zu verstehen und in analytische Fragestellungen zu überführen, geeignete Datenquellen zu identifizieren und die Qualität von Daten zu bewerten. Behandelt werden grundlegende statistische Methoden, Datenaufbereitung, einfache Analyse- und Modellierungsansätze sowie die Transformation und Verarbeitung gängiger Datenformate.

Ein weiterer Schwerpunkt liegt auf Datenvisualisierung und Data Storytelling, einschließlich der Interpretation von Ergebnissen, der Ableitung von Hypothesen und der Formulierung datenbasierter Handlungsempfehlungen. Ergänzend erfolgt eine Einführung in die Programmiersprachen R und Python auf Anfängerniveau sowie in grundlegende Werkzeuge der Business Intelligence.

Das Modul integriert Inhalte aus dem Bereich Data Literacy und legt besonderen Wert auf ethische Aspekte, Datenqualität und den verantwortungsvollen Umgang mit Daten. Die Ergebnisse werden im Rahmen von Präsentationen dokumentiert und vorgestellt.

Lernergebnisse:

Ziel des Moduls ist es, ein grundlegendes Verständnis für datengetriebene Entscheidungsprozesse zu entwickeln. Die Studierenden sind in der Lage, einfache Datenanalysen durchzuführen, Ergebnisse zu interpretieren und diese verständlich zu kommunizieren.

Nach Abschluss des Moduls können die Teilnehmer eine konkrete Problemstellung in einen datenbasierten Analyseprozess überführen, einfache Modelle zur Entscheidungsunterstützung entwickeln und Daten mithilfe grundlegender statistischer und visueller Methoden auswerten. Darüber hinaus erwerben sie Erfahrung in Teamarbeit, Selbstorganisation und Ergebnispräsentation.

Voraussetzungen und Empfehlungen:

Vorausgesetzt werden grundlegende mathematische Kenntnisse, wie beispielweise der Bruchrechnung und einfache Logik, sowie ein Basisverständnis von Wertschöpfungsprozessen in Unternehmen. Programmierkenntnisse sind nicht erforderlich.

Literatur:

Data Literacy: How to Make Your Experiments Robust and Reproducible; Academic Press 2017; Neil Smalheiser

Data Literacy: A User's Guide; SAGE Publications 2015; David Herzog

The Basics of Data Literacy; National Science Teachers Association 2014; Michael Bowen, Anthony Bartley

Data Analytics for Absolute Beginners: A Deconstructed Guide to Data Literacy; Independently published 2019; Oliver Theobald

Introduction to Robotics; Addison Wesley 1991; Phillip John McKerrow

Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking; O'Reilly Media 2013; Foster Provost, Tom Fawcett

The Truthful Art: Data, Charts, and Maps for Communication; New Riders 2016; Alberto Cairo

The Signal and the Noise: Why So Many Predictions Fail – but Some Don't; Penguin 2013; Nate Silver

Naked Statistics: Stripping the Dread from the Data; Brilliance Corp 2014; Charles Wheelan

The Art of Statistics: Learning from Data; Pelican Books 2019; David Spiegelhalter

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science

◆ MB369 – Digitaltechnik

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB069 – Prakt. Digitaltechnik	Teilleistung:	TB364 – Digitaltechnik
Lernform:	Praktikum	Lernform:	Vorlesung
Prüfungsform:	Praktikumsbericht / Protokoll	Prüfungsform:	Klausur
Prüfungsdauer:	30 Min.	Prüfungsdauer:	90 Min.
ECTS:	2,0	ECTS:	3,0
Benotung:	Bestanden/nicht Bestanden	Benotung:	Drittelnoten
Turnus:	jedes Semester	Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden	Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	4 Termine	Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	10 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	4 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden	Aufwand während Semesterferien:	5 Stunden
Flexibel einteilbarer Aufwand:	46 Stunden	Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	60 Stunden	Gesamtaufwand:	<i>Gesamtaufwand 75 Stunden weicht signifikant von erwarteten 3,0 ECTS ab.</i>
Lehrende:	Thomas Starke	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
- Schaltungsnetzwerk: Schaltwerk 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:	Gesamtaufwand 105 Stunden weicht signifikant von erwarteten 5,0 ECTS ab.
Lehrende:	Sergei Sawitzki Dennis Strieg

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	Teilleistung:	TB383 – Übg. Objektorientierte Programmstrukturen
Lernform:	Vorlesung	Lernform:	Übung
Prüfungsform:	Klausur + ggf. Bonus	Prüfungsform:	Abnahme
Prüfungsdauer:	120 Min.	Prüfungsdauer:	30 Min.
ECTS:	2,0	ECTS:	3,0
Benotung:	Drittelnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester	Turnus:	jedes Semester
Dauer (pro Termin):	8 Semesterwochenstunden	Dauer (pro Termin):	6 Semesterwochenstunden
Termine im Semester:	6 Termine	Termine im Semester:	6 Termine
Zeit in Veranstaltungen:	30 Stunden	Zeit in Veranstaltungen:	22 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	15 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	50 Stunden
Aufwand während Semesterferien:	15 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	10 Stunden
Gesamtaufwand:	60 Stunden	Gesamtaufwand:	82 Stunden
Lehrende:	Dennis Proppe	Lehrende:	Gert 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 Kopeck: 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:	1,50
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 Siring 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:	MB377 – 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:	Gesamtaufwand 135 Stunden weicht signifikant von erwarteten 5.0 ECTS ab
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) einteilen 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

◆ MB040 – Algorithmen und Datenstrukturen

Verantwortliche:	Christian Uhlig
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB015 – Algorithmen und Datenstrukturen	Teilleistung:	TB016 – Übg. Algorithmen und Datenstrukturen
Lernform:	Vorlesung	Lernform:	Übung
Prüfungsform:	Klausur	Prüfungsform:	Abnahme
Prüfungsdauer:	90 Min.	Prüfungsdauer:	75 Min.
ECTS:	2,0	ECTS:	2,0
Benotung:	Drittelnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester	Turnus:	Wintersemester
Dauer (pro Termin):	4 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	36 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	57 Stunden
Aufwand während Semesterferien:	24 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden	Gesamtaufwand:	Gesamtaufwand 72 Stunden weicht signifikant von erwarteten 2,0 ECTS ab.
Lehrende:	Christian Uhlig	Lehrende:	Malte Heims

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 Sortieralgorithmen 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	Teilleistung:	TB074 – Übg. Systemnahe Programmierung
Lernform:	Vorlesung	Lernform:	Übung
Prüfungsform:	Klausur	Prüfungsform:	Abnahme
Prüfungsdauer:	120 Min.	Prüfungsdauer:	35 Min.
ECTS:	2,0	ECTS:	3,0
Benotung:	Drittelnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester	Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	25 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	67 Stunden
Aufwand während Semesterferien:	20 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden	Gesamtaufwand:	<i>Gesamtaufwand 102 Stunden weicht signifikant von erwarteten 3,0 ECTS ab.</i>
Lehrende:	Christian Uhlig	Lehrende:	Malte Heims

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

◆ MB116 – Technologie der Mediengestaltung und GUI-Programmierung

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB009 – Technologie der Mediengestaltung und GUI-Programmierung
Lernform:	Vorlesung
Prüfungsform:	Portfolio-Prüfung
Prüfungsdauer:	60 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:	90 Stunden
Aufwand während Semesterferien:	30 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Ole Nass

Studieninhalte:

- Motivation, Begriffe und Konzepte
 - Mensch-Computer-Interaktion (MCI): Ziele, Herausforderungen, Modelle
 - Überblick, Technologien webbasierter Programmierung
- Menschliche Informationsverarbeitung und ihre Bedeutung für die MCI
 - Modelle zur Informationsverarbeitung
 - Sinne und ihre Relevanz
 - Wahrnehmungsgesetze und Gedächtnis
 - Handlungspsychologie und das Interface als Handlungsraum
 - Handlungsprozesse und Fehlerbehandlung
- Interaktion im Dialog
 - Funktions- und ablauforientierte Interaktion
 - Gestaltungsgrundsätze
 - Wahrnehmungsbasierte Organisation komplexer Informationen
 - Navigation in multimedialen Anwendungen
 - Normen, Gesetze, Richtlinien
 - Barrierefreiheit
- Technologien für Dynamische Webseiten
 - Client-Server-Modell
 - Frameworks
 - Begriffe: Mandantenfähigkeit, Backend, Backoffice, Frontend, Template
- Konkrete Technologien für unterschiedliche Clients
 - Auswahl aktueller Technologien
 - Übungen: z. B. HTML5, CSS, AJAX, JavaScript, Webapp, Flash, Flex / Air...

Lernergebnisse:

Studierende erhalten ...

- die Fähigkeit, aktuelle Technologien der Programmierung von webbasierten Medien funktional und operativ zu durchdringen,
- die Fähigkeit, die Aspekte, mit denen spezielle Gestaltungsvorstellungen umzusetzen sind, zu erkennen,
- das Verständnis der physiologischen und psychologischen Grundkonzepte von Interaktionen, das die Zusammenhänge zwischen menschlicher Informationsverarbeitung und Konzepten zur Analyse und Gestaltung interaktiver Systeme transparent macht,
- das Verständnis der softwareergonomischen Richtlinien / Normen zu den Informationstechnik-Verordnungen zur Barrierefreiheit sowie
- das Verständnis der unterschiedlichen Hardwarekonzepte für interaktive Ein- und Ausgabemedien.

Voraussetzungen und Empfehlungen:

- Vertrautheit mit webbasierten Technologien wie HTML5, CSS und JavaScript.
- Fähigkeit zur Entwicklung dynamischer Webseiten und Anwendungen.

Literatur:

- Böhringer, J., Bühler, P., Schlaich, P., Sinner, D. (2014): Kompendium der Mediengestaltung, 6. Auflage, Berlin: Springer Verlag (4 Bände: Konzeption, Technik, Print, Digital)
- Butz, A., Krüger, A. (2014): Mensch-Maschine-Interaktion. München: De Gruyter/ Oldenbourg
- Dahm, M. (2006): Grundlagen der Mensch-Computer-Interaktion. München: Pearson Studium
- Hammer, N., Bensmann, K. (2011): Webdesign für Studium und Beruf. Webseiten planen, gestalten und umsetzen. 2. Aufl. Berlin: Springer Verlag.
- Heinecke, A.M. (2012): Mensch-Computer-Interaktion. 2. Aufl. Berlin: Springer
- Khazaeli, C.D. (2005): Systemisches Design, Intelligente Oberflächen für Information und Interaktion. Reinbek: Rowohlt Verlag
- Malaka, R., Butz, A., Hußmann, H. (2009): Medieninformatik. Eine Einführung. München: Pearson Studium
- Preim, B., Dachselt, R. (2015): Interaktive Systeme, Band 1+2. Berlin: Springer Vieweg
- Sharp, Helen; Rogers, Yvonne; Preece, Jenny (2007): Interaction Design: Beyond Human-Computer Interaction. 2nd edition, New York: Wiley.
- Stephanidis, Constantine (ed) (2009): Universal Access in Human Computer Interaction. Intelligent and Ubiquitous Interaction Environments. 5th International Conference UAHCI 2009, San Diego, CA, USA (LNCS 5615). Berlin, New York: Springer

Studiengänge:

- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- Medieninformatik 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:	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:	30 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	Gesamtaufwand 45 Stunden weicht signifikant von erwarteten 3.0 ECTS ab
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
-

◆ MB372 – Künstliche Intelligenz

Verantwortliche:	Sebastian Iwanowski
Moduldauer:	6 Monate
Unterrichtssprache:	None

Bestandteile:

Teilleistung:	TB367 – Künstliche Intelligenz
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	120 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:	60 Stunden
Flexibel einstellbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Hendrik Annuth Dennis Siring Sebastian Iwanowski

Studieninhalte:

- Einführung
 - Definition und Ziele der KI
 - Überblick über die Basistechnologien der KI
 - Auswahl von Anwendungsbeispielen
- Basistechnologien
 - Wissensbasierte Systeme mit den Ausprägungen Regelbasierte Systeme und Modellbasierte Systeme, Abgrenzung zum Machine Learning
 - Suchstrategien
 - Schwarmintelligenz
 - CNN Architecture (Summary)
 - Recurrent Neural Network RNNs (incl. LSTM)
 - Transformer Architecture (Vision Transformer)
- Anwendungen
 - Verkehrsinformation und -navigation
 - Logistische Fragestellungen
 - Technische Diagnose

Lernergebnisse:

Nach Abschluss der Veranstaltung besitzen die Studierenden folgende Kompetenzen:

- Kenntnis und Interesse für die grundsätzlichen Ziele der Künstlichen Intelligenz.
- Kenntnis der Basistechnologien der Künstlichen Intelligenz.
- Fähigkeit, elementare Techniken der Künstlichen Intelligenz in Implementierungen anzuwenden.
- Kenntnis verschiedener komplexer Anwendungsbeispiele.

Voraussetzungen und Empfehlungen:

Gute Programmierkenntnisse, Diskrete Mathematik, etwas Statistik

Literatur:

- Marco Dorigo / Thomas Stützle:
Ant Colony Optimization,
MIT Press 2004, ISBN 0-262-04219-3
- Goodfellow, Ian, Yoshua Bengio und Aaron Courville: Deep Learning. <http://www.deeplearningbook.org>. MIT Press, 2016. ISBN: 978-0-262-03561-3
- Ute Schmid / Günter Görz / Josef Schneeberger (Hsg.):
Handbuch der Künstlichen Intelligenz,
Oldenbourg 2013 (5. Auflage), ISBN 978-3-486-71307-7
- Tanya Braun / Eike Hüllermeier / Ute Schmid / Günter Görz:
Handbuch der Künstlichen Intelligenz,
Oldenbourg 2026 (7. Auflage), ISBN 978-3-11-138649-2
- Stuart Russell / Peter Norvig:
Künstliche Intelligenz: Ein moderner Ansatz,
Pearson Studium 2022 (4. Auflage), ISBN 978-386-894430-3

Studiengänge:

- Computer Games Technology Bachelor of Science
- Informatik Bachelor of Science
- IT-Ingenieurwesen Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Medieninformatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen 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 einstellbarer 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	Teilleistung:	TB014 – Prakt. Rechnernetze
Lernform:	Vorlesung	Lernform:	Praktikum
Prüfungsform:	Klausur	Prüfungsform:	Absnahme
Prüfungsdauer:	90 Min.	Prüfungsdauer:	2,0
ECTS:	2,0	ECTS:	2,0
Benotung:	Drittelnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester	Turnus:	jedes Semester
Dauer (pro Termin):	4 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	60 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden	Gesamtaufwand:	60 Stunden
Lehrende:	Ilja Kaleck	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)
 - Schwerpunktbehandlung: 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" Simulators

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 Koppellemente 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 zum 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 978-3-44621501-6
- 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:	Gesamtaufwand 101 Stunden weicht signifikant von erwarteten 3,0 ECTS ab.
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

◆ MB059 – Web-Anwendungen

Verantwortliche:	Marian Gajda
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB027 – Web-Anwendungen	Teilleistung:	TB028 – Übg. Web-Anwendungen
Lernform:	Vorlesung	Lernform:	Übung
Prüfungsform:	Klausur	Prüfungsform:	Abnahme
Prüfungsdauer:	60 Min.	Prüfungsdauer:	45 Min.
ECTS:	2,0	ECTS:	2,0
Benotung:	Drittelnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester	Turnus:	Sommersemester
Dauer (pro Termin):	3 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	22 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	40 Stunden
Aufwand während Semesterferien:	0 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	82 Stunden	Gesamtaufwand:	55 Stunden
Lehrende:	Marian Gajda	Lehrende:	Marian Gajda

Studieninhalte:

- Basiskonzepte des WWW
 - Klassische Auszeichnungsmöglichkeiten in HTML
 - HTML-Formulare und ihre Möglichkeiten
 - Style Sheets
 - CSS-Animationen
 - Templating
 - Responsive Design
- Dynamik in Web-Seiten mit Javascript
 - Client-seitige Dynamik
 - Server-seitige Dynamik
- Asynchronous Javascript

Bearbeitung von Übungsaufgaben, die sich am Stoff der Vorlesung orientieren, in Zweiergruppen mit Abnahme der Lösungen. Erstellt wird eine im Verlaufe der einzelnen Übungseinheiten komplexer werdende Web-Anwendung, wobei die einzelnen Schritte aufeinander aufbauen, so dass am Ende eine komplexe Web-Anwendung entsteht, die einen Großteil der in der Vorlesung erlernten Techniken und Konzepte nutzt.

Lernergebnisse:

Die Studierenden ...

- sind in der Lage, die in der Vorlesung vermittelten theoretischen Hintergründe selbst praktisch anzuwenden.
- haben umfangreiche Kenntnisse und praktische Erfahrungen zu den Themen HTML, CSS, serverseitiger Dynamik, clientseitiger Dynamik mit JavaScript und AJAX, Einsatz JSON zum Austausch von Daten zwischen Client und Server, Einsatz von Cookies und Sessions zum temporären Speichern von Daten.
- steigern ihre Teamfähigkeit durch intensive Arbeit in Zweierteams und Kommunikation über auftretende Probleme in der ganzen Gruppe.

Die Studierenden ...

- führen die technischen Randbedingungen des Internet auf und benennen ihre Auswirkungen.
- beschreiben die konzeptionellen Aspekte von Stylesheets und der zentralen Möglichkeiten zur Festlegung der Darstellung in den Cascading Stylesheets und nutzen diese zur Erzeugung angestrebter Darstellungsweisen.
- können responsive Web-Layouts erstellen
- kennen wichtige Konzepte, Sprachen, Frameworks und Architekturen zur Realisierung dynamischer Webseiten auf, wählen zwischen diesen problembezogen aus und nutzen sie zur Erstellung dynamischer Webseiten.
- geben die zusätzliche Konzepte und Sprachelemente von HTML 5 an und entwerfen damit Webseiten.
- nutzen die theoretisch vermittelten Inhalte zur eigenständigen Realisierung von Webanwendungen begrenzter Komplexität.

Voraussetzungen und Empfehlungen:

Die Studierenden müssen gegebenen Quelltext lesen und fehlerfrei reproduzieren können. Die in vorigen Veranstaltungen erlernten Fertigkeiten im Umgang mit einem Versionskontrollsystem wird in der Übung vorausgesetzt. Es ist hilfreich, wenn die Grundlagen der Netzwerktechnik (Hostname, IP-Adresse, Port) bekannt sind.

Zur Vorbereitung auf Heimarbeit kann ein beliebiger Quelltext-Editor installiert werden.

Literatur:

- WOLF, Jürgen: HTML5 und CSS - Das umfassende Handbuch 2019
- ACKERMANN, Philipp: JavaScript - Das umfassende Handbuch 2019
- LABORENZ, Kai: CSS: Das umfassende Handbuch. Galileo Computing, 2011
- GASSTON, Peter: Moderne Webentwicklung: Geräteunabhängige Entwicklung - Techniken und Trends in HTML5, CSS3 und JavaScript, dpunkt.verlag, 2014
- WORLD WIDE WEB CONSORTIUM: HTML 5. <http://www.w3.org/TR/2014/WD-html5-20140617/>

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
-

◆ MB097 – Bildbearbeitung und -analyse

Verantwortliche:	Dennis Säring
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB076 – Bildbearbeitung und -analyse	Teilleistung:	TB093 – Prakt. Bildbearbeitung und -analyse
Lernform:	Vorlesung	Lernform:	Übung
Prüfungsform:	Klausur	Prüfungsform:	Abnahme
Prüfungsdauer:	60 Min.	Prüfungsdauer:	
ECTS:	2,0	ECTS:	3,0
Benotung:	Drittelnoten	Benotung:	Drittelnoten
Turnus:	Sommersemester	Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden
Aufwand während Semesterferien:	0 Stunden	Aufwand während Semesterferien:	16 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden	Gesamtaufwand:	91 Stunden
Lehrende:	Dennis Säring	Lehrende:	Herrmann Höhne

Studieninhalte:

- Einführung in die Bildbearbeitung
- Visualisierung und Bildanpassung
- Komposition und Filterung
- Fourier-Transformation und Frequenzfilter
- Lineare und nicht-lineare Registrierung
- Segmentierung und Texturanalyse
- Klassifikationsverfahren
- Selbstständiges Einarbeiten in den C++17 Standard auf Basis des vorhandenen Hintergrundwissens um C
- Selbstständiges Einarbeiten in die Basisfunktionen von OpenCV bei minimaler Hilfestellung
- Implementierung von Algorithmen zur Bildbearbeitung und -analyse:
 - Punktweise Transformationen (Fensterung, Gamma-Korrektur)
 - Lineare Faltungsfilter und morphologische Filter
 - Affine Transformationen und Interpolation
 - Farb- und Texturanalyse
 - Grundlagen der Objekterkennung inklusive Segmentierung und Formerkennung

Lernergebnisse:

Die Studierenden ...

- haben ein Verständnis für die Struktur digitaler Bilddaten
- erkennen die Möglichkeiten der Darstellung und Anpassung von digitalen Bildern
- können das Konzept der Bildkomposition und der Bildfilterung in praktischen Übungen umsetzen
- sind in der Lage Bilddaten in Frequenz- und Ortsbereich zu analysieren und zu bearbeiten
- verfügen über die theoretischen Grundlagen zur Registrierung von Bilddaten
- kennen unterschiedliche Ansätze Objekte in Bilddaten automatisch zu klassifizieren und zu segmentieren
- sammeln im Praktikum Erfahrungen beim Implementieren von Algorithmen zur Bildbearbeitung
- vertiefen durch praktisches Umsetzen die in der zugehörigen Vorlesung theoretisch erläuterten Algorithmen zur Bildbearbeitung und -analyse
- üben das selbstständige Erweitern ihrer Kenntnisse in Programmiersprachen
- üben das selbstständige Einarbeiten in eine vorgegebene Bibliothek
- können diese Fähigkeiten im kommerziellen oder akademischen Umfeld praktisch nutzen

Voraussetzungen und Empfehlungen:

Zur Vorbereitung auf das Praktikum kann auf dem heimischen Rechner eine Entwicklungsumgebung für C++ installiert werden. Das in den vorigen Veranstaltungen erworbene Wissen über die Syntax von C und die objektorientierten Konzepte von Java können zwecks Kombination wiederholt werden.

Literatur:

- Burger, Burge: Digitale Bildverarbeitung, Springer-Verlag 2005
- Handels: Medizinische Bildverarbeitung, Vieweg+Teubner 2009
- Howse, Minichino: Learning OpenCV 4 Computer Vision with Python 3, Packt Publishing 2020

Studiengänge:

- Computer Games Technology Bachelor of Science
- Informatik Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Technische Informatik 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:	Drittnoten
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, Kohn; 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

◆ MB223 – Machine Learning

Verantwortliche:	Hendrik Annuth
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB111 – Machine Learning	Teilleistung:	TB122 – Übg. Machine Learning
Lernform:	Vorlesung	Lernform:	Vorlesung
Prüfungsform:	Klausur	Prüfungsform:	Absnahme
Prüfungsdauer:	90 Min.	Prüfungsdauer:	120 Min.
ECTS:	2,0	ECTS:	3,0
Benotung:	Drittelnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester	Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	40 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	75 Stunden
Aufwand während Semesterferien:	0 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	55 Stunden	Gesamtaufwand:	90 Stunden
Lehrende:	Hendrik Annuth	Lehrende:	Hendrik Annuth

Studieninhalte:

- Einlesen, Bereinigen, Normalisieren und Erweitern von Datensätzen
- Durchführung von Regression, Classification und Auto Classifications Aufgaben
- Ergebnisoptimierung durch die Einstellung von Parametern innerhalb des Lernvorgangs
- Lerntypen: supervised, unsupervised, reinforcement, semi-supervised
- Aufgabentype: Regression, Classification, Auto Classification, Colaborative Filtering
- Überblick über die wichtigsten Metriken
- Datentypen und Verfahren: NLP, Computer Vision, Tabular Data
- Under- und Overfitting
- Gradient Decent
- Vanishing- und Exploding-Gradient-Problem
- Single Layered Perception, Multilayered Perception
- Learning Rate: Momentum, One Cycle Learning
- Optimierungen: SGD, Adam, RMSprop, Drop-Out, Batch Normalization, Weight Decay
- Convolutional Neural Networks
- Recurrent Neural Networks
- Common Machine Learning Libraries

Lernergebnisse:

- Welche Voraussetzungen haben Machine-Learning-Verfahren, wann, wie und wo ist deren Einsatz sinnvoll?
- Ein detailliertes Verständnis für die Vor- und Nachteile von Machine-Learning-Prozessen
- Das Gradientenabstiegsverfahren wird so detailliert behandelt, dass auch Änderungen und Erweiterungen des Verfahrens eingeordnet und verstanden werden können
- Der gekonnte Einsatz von Optimierungstechniken innerhalb des Lernprozesses
- Die Fähigkeit, Under- und Overfitting an deren jeweiligen Auswirkungen zu erkennen und sinnvolle Gegenmaßnahmen zu ergreifen
- Für verschiedene Problemstellungen, wie Bilderkennung, Textvervollständigung, Empfehlungssysteme und Klassifikationsprobleme eine sinnvolle künstliche neuronale Netzwerkarchitektur auswählen zu können
- Sicherer Umgang mit der Aufbereitung von Datensätzen für die Benutzung von Machine Learning
- Erfahrung mit der richtigen Auswahl einer künstlichen neuronalen Netzarchitektur für ein gegebenes Ziel mit gegebenen Daten
- Ein gutes Verständnis für die Auswahl und Einstellung von Optimierungsparametern zum Zwecke der Ergebnisoptimierung
- Erfahrung mit der Auswahl, Auswirkung und Bedeutung von verschiedenen Metriken

Voraussetzungen und Empfehlungen:

Das Modul Exploratory Data Analysis sind wichtige Voraussetzungen für das Modul. Erweitertes mathematisches Wissen aus dem Gebiet Statistik und Lineare Algebra wird in der Veranstaltung vorausgesetzt. Zusätzlich werden Fähigkeiten zur Programmierung in R und Python vorausgesetzt.

Literatur:

- The Elements of Statistical Learning; Springer 2009; Trevor Hastie, Robert Tibshirani, Jerome Friedman
- Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython; O'Reilly and Associates 2012
- Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow; O'Reilly Media 2019; Aurélien Géron
- An Introduction to Statistical Learning; Springer 2013; Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani
- Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow; Packt Publishing 2019; Sebastian Raschka, Vahid Mirjalili
- Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play; O'Reilly UK Ltd. 2019; David Foster

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Smart Technology Bachelor of Science

◆ MB232 – Formale Sprachen

Verantwortliche:	Marco Pawlowski
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB057 – Formale Sprachen
Lernform:	Vorlesung
Prüfungsform:	Klausur
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:	60 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	Gesamtaufwand 90 Stunden weicht signifikant von erwarteten 3,0 ECTS ab
Lehrende:	Marco Pawlowski

Studieninhalte:

- Endliche Automaten
 - Endliche Automaten mit und ohne Ausgabe
 - Deterministische und nichtdeterministische Automaten
 - Reguläre Sprachen
 - Reguläre Ausdrücke
 - Äquivalenz und Minimierung endlicher Automaten
 - Anwendungen endlicher Automaten
 - Pumping-Lemma für reguläre Sprachen
 - Nicht-reguläre Sprachen
- Formale Sprachen
 - Ersetzungssysteme
 - Grammatiken
 - Chomsky-Hierarchie
 - Kontextfreie Grammatiken
 - Normalformen
 - Pumping-Lemma für kontextfreie Sprachen
 - Abschlusseigenschaften kontextfreier Sprachen
 - Entscheidbarkeit bei kontextfreien Sprachen
 - Eindeutigkeit und Mehrdeutigkeit von Grammatiken
 - Syntaxanalyse von Programmiersprachen
 - Kellerautomaten
 - Kontext-sensitive Sprachen
 - Linear-beschränkte Automaten
 - Turingmaschinen
- Compiler im Überblick
 - Compilerphasen
 - Compiler und Interpretierer
 - Anwendungsfelder
- Lexikalische Analyse
 - Nutzung regulärer Ausdrücke und nicht-deterministischer bzw. deterministischer endlicher Automaten
 - Scanner und Scanner-Generatoren
- Syntaxanalyse
 - Rekursiver Abstieg
 - LL- und LR-Parser
 - Parser-Generatoren
- Semantische Analyse
 - Typüberprüfung
- Codeerzeugung und -optimierung
- Virtuelle Maschinen

Lernergebnisse:

Die Studierenden ...

- unterscheiden verschiedene Arten von Grammatiken und Sprachklassen.
- nennen die Eigenschaften abstrakter Automaten.
- erkennen den Zusammenhang zwischen Automaten und Sprachen.
- nennen die Grundbegriffe der Programmsyntax und von Compilern.
- beschreiben den Übersetzungsprozess formaler Sprachen.
- nennen und charakterisieren die Phasen des Übersetzungsprozesses.
- entwerfen einfache domänenspezifische formale Sprachen und definieren sie mittels regulärer Ausdrücke und kontextfreier Grammatiken.
- nutzen Scanner- und Parsergeneratoren zur Verarbeitung einfacher domänenspezifischer Sprachen.
- erläutern die Transformation höherer Programmiersprachen in Maschinensprache.
- verstehen und differenzieren die Arbeitsweise von Compilern und Interpretierern.
- verstehen Funktionsweise und Anwendungsbereiche virtueller Maschinen.

Voraussetzungen und Empfehlungen:

Konzepte der Diskreten Mathematik, dabei unter anderem die Fähigkeit mathematische Formeln zu verstehen und den Aufbau von Programmiersprachen zu kennen ist empfehlenswert.

Literatur:

- Aho, Sethi und Ullman: Compiler Prinzipien, Techniken und Werkzeuge, ISBN: 978-3-86326-574-8
- John E. Hopcroft / Rajeev Motwani / Jeffrey D. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie. 2. überarb. Aufl. München: Addison-Wesley Longman Verlag, 2002, ISBN 978-3827370204

Studiengänge:

- Computer Games Technology Bachelor of Science
- Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB253 – Geschäftsprozesse mit ERP-Systemen

Verantwortliche:	Fikret Koyuncu
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB059 – Geschäftsprozesse mit ERP-Systemen	Teilleistung:	TB060 – Übg. Geschäftsprozesse mit ERP-Systemen
Lernform:	Vorlesung	Lernform:	Übung
Prüfungsform:	Klausur	Prüfungsform:	Abnahme
Prüfungsdauer:	90 Min.	Prüfungsdauer:	15 Min.
ECTS:	2,0	ECTS:	3,0
Benotung:	Drittnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester	Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden	Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden	Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden
Aufwand während Semesterferien:	0 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden	Gesamtaufwand:	90 Stunden
Lehrende:	Fikret Koyuncu	Lehrende:	Fikret Koyuncu

Studieninhalte:

Die Veranstaltung soll den Studierenden wichtige Grundlagen zum Verständnis und der Anwendung von ERP-Systemen in der Unternehmenspraxis vermitteln. Hierzu werden neben den begrifflichen Grundlagen und Inhalten zur Systemarchitektur von ERP-Systemen die wesentlichen Konzepte zur Abbildung von Geschäftsprozessen in ERP-Systemen vermittelt. Schwerpunkt bilden hierbei die Themen Daten- und Funktionsintegration aus einer prozessorientierten Perspektive.

Kurzgliederung:

- Grundlagen und Begriffe
- ERP-System-Architektur
- Integration von Geschäftsprozessen und ERP-Systemen
- Prozesse in der Produktion und deren Abbildung in ERP-Systemen
- Prozesse im Bereich Materialwirtschaft, Logistik und deren Abbildung in ERP-Systemen
- Prozesse im Bereich Controlling und deren Abbildung in ERP-Systemen
- Prozesse im Bereich Rechnungswesen und deren Abbildung in ERP-Systemen

Die in der Vorlesung vermittelten theoretischen Grundlagen werden im Rahmen der Übung durch die Bearbeitung von Fallstudien durch eigene Anwendung gefestigt und erweitert. Die Bearbeitung der durchgängigen und ERP-modulübergreifenden Fallstudien erfolgt direkt am ERP-System am Beispiel des SAP ERP. Hiermit soll den Studierenden insbesondere eine integrierte Betrachtungsweise der betriebswirtschaftlichen Vorgänge und deren informationstechnische Abbildung in einem ERP-System anhand praktischer Übungen näher gebracht werden.

Kurzgliederung:

- Navigation im SAP ERP-System
- Fallstudie zu den Prozessen in der Produktion
- Fallstudie zu den Prozessen im Bereich Materialwirtschaft, Logistik
- Fallstudie zu den Prozessen im Bereich Controlling
- Zusammenhängendes Fallbeispiel zu den behandelten Themen

Lernergebnisse:

Die Studierenden können ...

- die erworbenen theoretischen Kenntnisse auf praktische Anwendungsfälle übertragen
- die grundlegende Funktionen von ERP-Systemen verwenden
- betriebswirtschaftliche Aufgabenstellungen und zusammenhängende Geschäftsvorfälle in verschiedenen Problemfeldern mit Hilfe von ERP-Systemen bearbeiten.

Die Studierenden können ...

- wesentliche Begriffe im Umfeld von ERP-Systemen und Geschäftsprozessen definieren und diese in Beziehung zueinander setzen
- die wesentlichen Eigenschaften von ERP-Systemen und deren Architekturen herausstellen
- können die Vorgehensweise zur Abbildung betriebswirtschaftlicher Prozesse in ERP-Systemen erläutern.

Voraussetzungen und Empfehlungen:

keine

Literatur:

- Forsthuber, Heinz, Siebert, Jörg: Praxishandbuch SAP-Finanzwesen. Bonn; Boston, Mass : Galileo Press, 2013
- Frick, Detlev ; Gadatsch, A. ; Schäffer-Külz, U. G. (Hrsg.): Grundkurs SAP ERP: Geschäftsprozessorientierte Einführung mit durchgehendem Fallbeispiel. 2., Aufl. 2013. Aufl. Wiesbaden : Springer Fachmedien Wiesbaden GmbH, 2013
- Gadatsch, Andreas: Grundkurs Geschäftsprozess-Management: Methoden und Werkzeuge für die IT-Praxis: eine Einführung für Studenten und Praktiker. Wiesbaden : Vieweg + Teubner, 2012
- Mertens, Peter: Integrierte Informationsverarbeitung 1 Operative Systeme in der Industrie. Wiesbaden : Springer Fachmedien Wiesbaden: Imprint: Springer Gabler, 2013
- Weske, Mathias: Business process management concepts, languages, architectures. Berlin; New York : Springer, 2012

KOYUNCU, Fikret: Übungsmaterialien zu den jeweiligen Fallstudien auf Moodle

Studiengänge:

- Betriebswirtschaftslehre Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB305 – Datenbanktheorie und -implementierung

Verantwortliche:	Ulrich Hoffmann
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB296 – Datenbanktheorie und -implementierung	Teilleistung:	TB297 – Workshop container- und serviceorientierte Architekturen
Lernform:	Vorlesung	Lernform:	Workshop
Prüfungsform:	Klausur	Prüfungsform:	Teilnahme
Prüfungsdauer:	60 Min.	Prüfungsdauer:	20 Min.
ECTS:	2,0	ECTS:	1,0
Benotung:	Drittnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester	Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden	Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	3 Termine
Zeit in Veranstaltungen:	30 Stunden	Zeit in Veranstaltungen:	7 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	40 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	10 Stunden
Aufwand während Semesterferien:	20 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	30 Stunden	Flexibel einteilbarer Aufwand:	12 Stunden
Gesamtaufwand:	120 Stunden	Gesamtaufwand:	29 Stunden
Lehrende:	Heiko Sturzheim	Lehrende:	Jan Sorgenfrei

Studieninhalte:

- Relationale Datenbanksysteme
 - Das 3 - Ebenen - Architekturkonzept
 - Transaktionskonzept
 - Relationale Operationen und Relationenalgebra
 - Komponenten eines DBMSs und Betrieb eines DBSs
- Aufbau eines Datenbankverwaltungssystems (DBMS)
 - Systementwurf und Schichtenmodell
 - Architektur eines DBMS
 - Komponenten eines DBMS
- Speichersystem
 - Speicherverwaltung
 - Systempufferverwaltung
- Speicherungsstrukturen und Zugriffspfade
 - Freispeicherverwaltung
 - Abbildung von Datensätzen in Seiten
 - Zugriffspfade
- Satzorientierte Datenbankschnittstelle
- Sicherung der Integrität in Datenbanken
 - Semantische Integrität
 - Operationale Integrität
 - Fehlerbehandlung - Recovery
- Grundlagen NoSQL-Datenbanken

Lernergebnisse:

Nach Besuch der Veranstaltungen dieses Moduls besitzen die Studierenden eingehende Kenntnisse der für die Implementierung von Datenbanksystemen wichtigen Architekturprinzipien, ihrer Datenstrukturen und Algorithmen. Sie besitzen die Fähigkeit Datenbanksysteme in Hinblick auf ihren Aufbau und der internen Funktionsweise zu bewerten und zu beurteilen. Sie sind in der Lage selbst Modifikationen an Datenbanksystem-Implementierungen zu planen und durchzuführen. Die Studierenden besitzen die Fähigkeit, die Arbeitsweise von Datenbanksystemen zu optimieren und auch selbst Architekturen für die Datenhaltung zu entwerfen. Sie besitzen grundlegende Kenntnisse der Administration von Datenbanken.

Die Studierenden besitzen ...

- Kenntnisse der relationalen Datenbanktheorie und ihrer wesentlichen Fragestellungen;
- detaillierte Kenntnisse der für die Implementierung von Datenbanksystemen wichtigen Architekturprinzipien, Datenstrukturen und Algorithmen und damit des Aufbau und der interne Arbeit eines großen komplexen Softwaresystems;
- die Fähigkeit, die Arbeitsweise von Datenbanksystemen zu optimieren bzw. selbst Architekturen für große komplexe Softwaresysteme zu entwerfen;
- Fähigkeit eines Datenbankadministrators für Datenbanksysteme.

Voraussetzungen und Empfehlungen:

Es werden grundlegende Kenntnisse in der Benutzung von Datenbanksystemen und Kenntnisse in SQL vorausgesetzt.

Literatur:

- Elmasri, Ramez; Navathe, Shamkant B.: Grundlagen von Datenbanksystemen. 3. Auflage. München: Pearson -Verlag, 2009.
- Kemper, Alfons; Eickler, André: Datenbanksysteme – Eine Einführung. 10. Auflage. München: De Gruyter Oldenbourg, 2015.
- Saake, Gunter; Sattler, Kai-Uwe; Heuer, Andreas: Datenbanken Implementierungstechniken. 3. Auflage. mitp-Verlag, 2011
- Vossen, Gottfried: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. 5. Auflage. München: Oldenbourg Verlag, 2008.

Studiengänge:

- Informatik Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB036 – Programmierpraktikum

Verantwortliche:	Gerit Kaleck
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB012 – Programmierpraktikum
Lernform:	Praktikum
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	45 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	0 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	0 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	20 Stunden
Flexibel eintellbarer Aufwand:	130 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Gerit Kaleck

Studieninhalte:

Im Rahmen der Veranstaltung lernen die Teilnehmer, ausgehend von einer problemorientierten Aufgabenstellung ein Java-Programm mittleren Umfangs zu erstellen. Das Projekt wird eigenständig strukturiert und modularisiert, passende Datenmodelle entwickelt und eine benutzungsgerechte Oberfläche entworfen. Im Rahmen der Aufgabenstellung wird die entstandene Software getestet und dokumentiert.

Lernergebnisse:

Die Studierenden ...

- können anhand einer problemorientierten Aufgabenstellung ein Softwareprojekt zeitlich und inhaltlich strukturieren und implementieren
- haben die Fähigkeit, die Basiskonzepte objektorientierter Programmiersprachen in Java in einem Softwareprojekt mittleren Umfangs sinnvoll einzusetzen
- wenden die Grundregeln benutzungsgerechter Oberflächengestaltung an
- sind in der Lage, eine Software zu testen und zu dokumentieren
- können eine Revisionsverwaltung nutzen

Voraussetzungen und Empfehlungen:

Die in der Vorlesung "Objektorientierte Programmstrukturen" gelehrteten Inhalte sollten sicher beherrscht werden und die zugehörige Übung bestanden worden sein. Eine Entwicklungsumgebung und bei Bedarf ein gesonderter Client für die Revisionsverwaltung sollten installiert sein. Die Regeln und Hilfestellungen für dieses Praktikum sollten im Voraus gelesen werden.

Literatur:

- ULLENBOOM, Christian (2025):
Java ist auch eine Insel
<https://openbook.rheinwerk-verlag.de/javainsel/>, letzter Zugriff: 30.03.2026
- EPPLE, Anton:
JavaFX 8: Grundlagen und fortgeschrittene Techniken
dpunkt Verlag, 2015
ISBN-13: 978-3864901690
- ZÖRNER, Stefan:
Softwarearchitekturen dokumentieren und kommunizieren: Entwürfe, Entscheidungen und Lösungen nachvollziehbar und wirkungsvoll festhalten
Carl Hanser Verlag, 2021 (3. Auflage)
ISBN-13: 978-3446469280

Studiengänge:

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

◆ MB041 – Induktive Statistik

Verantwortliche:	Franziska Bönte
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB017 – Induktive Statistik
Lernform:	Vorlesung mit integrierter Übung
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:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Franziska Bönte

Studieninhalte:

- Wahrscheinlichkeitsrechnung
 - Grundlagen
 - Zufallsvariable und Wahrscheinlichkeitsverteilungen
 - Kombinatorik
 - Ausgewählte diskrete Verteilungen
 - Ausgewählte stetige Verteilungen
 - Hauptsätze der Wahrscheinlichkeitstheorie und Statistik
 - Approximationsregeln
- Stichproben
- Schätzverfahren
 - Punktschätzung
 - Intervallschätzung
- Testverfahren
 - Parametrische Testverfahren
 - Verteilungstests

Lernergebnisse:

Die Studierenden sind befähigt, weiterführende statistische Methoden zur Lösung komplexer Problemstellungen nutzen und die erzielten Ergebnisse korrekt zu interpretieren.

Die Studierenden erlangen ...

- Verteilungsannahmen für unterschiedliche ökonomische und naturwissenschaftliche Fragestellungen sinnvoll zu tätigen
- die Fähigkeit, Schätzwerte für die Parameter einer Grundgesamtheit zu bestimmen.
- die Fähigkeit, selbständig statistische Tests im Rahmen betrieblicher Aufgabenstellungen zu planen und durchzuführen und die Ergebnisse korrekt anzugeben.
- die Fähigkeit, Intervallwahrscheinlichkeiten unter Verwendung der wichtigsten diskreten und stetigen Dichte- und Verteilungsfunktionen zu berechnen.
- die Fähigkeit, Werte einer Grundgesamtheit zu schätzen und Hypothesen über die Werte einer Grundgesamtheit zu testen.
- die Fähigkeit, mittels geeigneter Computerprogramme statistische Untersuchungen großer Datenmengen vorzunehmen.
- Kenntnisse hinsichtlich des Einsatzes von Testverfahren im Rahmen der statistischen Qualitätskontrolle anhand von Problemstellungen aus der Wirtschaft.
- die Fähigkeit, sowohl eine Zeitreihe zu analysieren und die Komponenten einer Zeitreihe zu berechnen als auch kurz- und langfristige Prognosen durchzuführen.
- die Fähigkeit, die Genauigkeit von Prognosen kritisch zu bewerten.

Voraussetzungen und Empfehlungen:

keine Voraussetzungen, empfohlen werden aber Kenntnisse der deskriptiven Statistik.

Literatur:

- Bleymüller, Josef: Statistik für Wirtschaftswissenschaftler; 16. Aufl.; s.l.; Verlag Franz Vahlen; 2012
- Griffiths, William E.; Hill, R. Carter; Judge, George G.: Learning and practicing econometrics; ; New York; John Wiley; 1993
- Hansen, Gerd: Methodenlehre der Statistik; ; München; Vahlen; 1974
- Hansmann, Karl-Werner: Kurzlehrbuch Prognoseverfahren; ; Wiesbaden; s.l.; Gabler Verlag; 1983
- Lippe, Peter Michael von der: Wirtschaftsstatistik; 3., neubearb. u. erw. Aufl.; Stuttgart; Fischer; 1985
- Mood, Alexander MacFarlane; Boes, Duane C.; Graybill, Franklin A.: Introduction to the theory of statistics; 3. ed., international ed., [reprint.]; Auckland; McGraw-Hill; 2009
- Rüger, Bernhard: Induktive Statistik; 2., überarb. Aufl., 2. Nachdr; München; Oldenbourg; 1995
- Schlittgen, Rainer; Streitberg, Bernd H. J.: Zeitreihenanalyse; 3. Aufl., durchges. u. verb; München; R. Oldenbourg; 1989
- Zuckarelli, Joachim: Statistik mit R; ; Heidelberg; O'Reilly; 2017
- Bourier, Günther: Beschreibende Statistik. 11. Auflage. Wiesbaden: Springer Gabler, 2013.
- Kobelt, Helmut; Steinhausen, Detlef: Wirtschaftsstatistik für Studium und Praxis. 7. Auflage. Stuttgart: Schäfer-Poeschel Verlag, 2006.
- Schwarze, Jochen: Grundlagen der Statistik Band 2 : Wahrscheinlichkeitsrechnung und induktive Statistik. 10. Auflage. Berlin: nwb Studium 2013.
- Toutenburg, Helge u., a.: Induktive Statistik : Eine Einführung mit R und SPSS. 4. Auflage. Berlin: Springer-Verlag 2008.

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

◆ MB087 – Systemmodellierung

Verantwortliche:	Dennis Proppe
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB032 – Systemanalyse
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	60 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:	15 Stunden
Aufwand während Semesterferien:	30 Stunden
Flexibel einteilbarer Aufwand:	30 Stunden
Gesamtaufwand:	90 Stunden
Lehrende:	Dennis Proppe

Teilleistung:	TB033 – Prozessmodellierung
Lernform:	Übung
Prüfungsform:	Absnahme
Prüfungsdauer:	
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:	45 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden
Lehrende:	Christian Uhlig

Studieninhalte:

- Grundbegriffe der Systemanalyse
 - Gegenstand und Zielsetzung im Unternehmensumfeld
 - Methodische Grundlagen
 - Systemaufnahme
 - Rahmenbedingungen und Techniken der Informationsgewinnung
 - Thematische Untersuchungsbereiche
 - ER-Datenmodellierung
 - Systemmodellierung
 - Ereignisgesteuerte Prozessketten zur Modellierung von Geschäftsprozessen
 - Modellelemente schlanker EPK-Modelle und ihre Nutzung
 - Modellelemente erweiterter EPK-Modelle und ihre Nutzung
 - Business Process Model and Notation BPMN
 - Ausgewählte Modellelemente
 - Beispielmodelle
-
- Grundlagen des Geschäftsprozessmanagement
 - Motivation
 - Begriffe
 - Einordnung der Geschäftsprozessmodellierung
 - Bezüge zur Systemanalyse und zum Software-Engineering
 - ARIS-Methode
 - Sichtenkonzept
 - Schichtenkonzept
 - Überblick über Modelltypen und ihre Vernetzung
 - ARIS-Softwaresystem
 - Modellierung der Aufbauorganisation
 - ER-Datenmodellierung
 - Funktionsmodellierung
 - Prozessmodellierung
 - Wertschöpfungsketten (WSK) und Prozesslandkarten
 - EPK
 - Vernetzung mit anderen ARIS-Sichten (Daten, Aufbauorganisation)
 - BPMN (Process und Collaboration Diagrams)
 - Praktische Aufgabenstellungen
 - Ausschnittsweise und formfreie Modellierung von Prozessen aus einem beispielhaften Fachkonzept
 - Modellierung des Datenmodells zu einem beispielhaften Fachkonzept (ERD)
 - Ausschnittsweise Modellierung von Prozessen zu einem beispielhaften Fachkonzept (WSK / EPK und BPMN)
 - Ganzheitliche Modellierung von Aufbauorganisation, Daten und Prozessen zu einer Fallstudie (Organigramm, ERD, WSK, BPMN)

Lernergebnisse:

Die Studierenden ...

- beurteilen die generellen Möglichkeiten und Grenzen von Systemanalysen, insbesondere in Bezug auf die Gestaltung betrieblicher Informationssysteme.
- unterscheiden die wesentlichen Techniken zur Informationsgewinnung in Unternehmen einschließlich ihrer Vor- und Nachteile, bewerten Techniken im Kontext einer konkreten Informationsgewinnung.
- führen eine methodisch fundierte Informationsgewinnung in einem überschaubaren Problemkontext durch.
- erklären wichtige Bestandteile und Schritte der Systemaufnahme als Vorphase zur Systemmodellierung, sie setzen ausgewählte Formalismen zur Dokumentation der Aufnahmeergebnisse ein.
- stellen die im Unternehmensumfeld praktisch relevanten methodischen Ansätze zur Systemmodellierung dar und beurteilen diese hinsichtlich ihrer Eignung für bestimmte Erkenntnisziele im Rahmen einer Systemanalyse.
- beschreiben die zu den methodischen Ansätzen gehörenden Modellnotationen und setzen diese angemessen zur Modellierung ein.
- nutzen die Modellierungsmittel zum Aufbau von Analysemodellen begrenzter Komplexität für betriebswirtschaftlich ausgerichtete Informationssysteme und diesbezügliche Problemstellungen.

Die Studierenden ...

- nennen und erläutern die theoretischen Grundlagen des Geschäftsprozessmanagement und seine Begriffswelt.
- nennen und erläutern die Grundlagen der Geschäftsprozessmodellierung sowie ihre Einordnung in das Geschäftsprozessmanagement und andere Themenbereiche wie Softwareengineering, Datenbanken und Systemanalyse.
- stellen die Grundlagen der ARIS-Methode dar.

- wenden wesentliche Modellierungskonzepte des ARIS-Softwaresystems (insbesondere Objekte und Kanten, Definitions- und Ausprägungsebenen sowie Hinterlegungen) an.
- nutzen das ARIS-Softwaresystem in seinen wesentlichen Bedienkonzepten und -elementen zur Erstellung von miteinander vernetzten Modellen.
- analysieren komplexe textuelle Fachkonzeptbeschreibungen und unterscheiden dabei Inhalte der verschiedenen ARIS-Modellierungssichten (Organisation, Daten, Leistungen, Funktionen, Steuerung).
- entwerfen und gestalten in ästhetisch ansprechender Weise Modelle zentraler Modelltypen (ER-Modell, EPK, WSK, BPMN Process und Collaboration, Organigramm) zu komplexen Fachkonzeptbeschreibungen.

Voraussetzungen und Empfehlungen:

- Grundverständnis der Informationstechnologie
- Kenntnis in objektorientierten Programmiersprache(n)
- Grundsätzliches Verständnis betriebswirtschaftlicher Zusammenhänge

Literatur:

- KRALLMANN, H.; BOBRIK, A.; LEVINA, O.: Systemanalyse im Unternehmen - Prozessorientierte Methoden der Wirtschaftsinformatik, Oldenbourg, 2013
 - ALLWEYER, Thomas: BPMN 2.0 - Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung. 2. Auflage, Books on Demand, 2020
 - GADATSCH, Andreas: Grundkurs Geschäftsprozess-Management: Analyse, Modellierung, Optimierung und Controlling von Prozessen. 9. Aufl., Springer Vieweg, 2020
 - RUPP, Chr.: Systemanalyse kompakt, Springer Verlag, 2013
 - HÄUSLEIN, A.: Systemanalyse. vde-Verlag, 2004
 - KRÜGER, J.; UHLIG, Ch.: Praxis der Geschäftsprozessmodellierung. VDE Verlag, 2009
 - SCHEER, A.-W.: Architektur integrierter Informationssysteme. Springer-Verlag, Berlin, 1991
 - OBJECT MANAGEMENT GROUP OMG: Business Process Model and Notation (BPMN), Version 2.0, URL: <http://www.omg.org/spec/BPMN/2.0>, 2011
 - FREUND, Jakob ; RÜCKER, Bernd: Praxishandbuch BPMN 2.0. 4. aktualisierte Auflage, Carl Hanser Verlag, 2014
 - OESTERREICH, B.: Analyse und Design mit UML 2.3: Objektorientierte Softwareentwicklung. Oldenbourg, 2009
 - KECHER, Chr.: UML 2: Das umfassende Handbuch. Galileo Computing, 2011
-
- Krüger, Jörg; Uhlig, Christian: Praxis der Geschäftsprozessmodellierung - ARIS erfolgreich anwenden, VDE Verlag, 2009
 - Lehmann, Frank: Integrierte Prozessmodellierung mit ARIS, dpunkt.verlag, 2007
 - Scheer, August-Wilhelm: ARIS Vom Geschäftsprozess zum Anwendungssystem, 4. Auflage, Springer, 2002
 - Scheer, August-Wilhelm: ARIS Modellierungsmethoden, Metamodelle, Anwendungen, 4. Auflage, Springer, 2001
 - OMG: Business Process Model and Notation (BPMN) Version 2.0, 2011
 - Software AG: ARIS-Dokumentation (Methodenhandbuch, Bedienhandbücher), jeweils aktuellste Fassung

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Smart Technology Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science
- Wirtschaftsingenieurwesen Bachelor of Science

◆ MB093 – Softwarequalität

Verantwortliche:	Gerd Beuster
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB034 – Softwarequalität
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
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:	120 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einstellbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Jochen Brummstein

Studieninhalte:

- Einführung und Motivation
 - Definition des Begriffs "Software-Qualität"
 - Bedeutung der Software-Qualität
- Merkmale der Software-Qualität
- Software-Maße und -Metriken
- Modelle der Software-Qualität
- Einschränkungen der Software-Qualität und ihre Gründe
- Software-Qualitätsmanagement
 - Aufgabenbereiche
 - Grundlegende Prinzipien
- Maßnahmen der Software-Qualitätssicherung
 - Konstruktive Maßnahmen
 - Prozessbezogene Maßnahmen
 - Produktbezogene Maßnahmen
- Analytische Maßnahmen
 - Statische Prüftechniken
 - Dynamische Prüftechniken
- Testen als Maßnahme der Qualitätssicherung
- Black-Box- und White-Box-Testing
- Verfahren des Black-Box-Testing
- Verfahren des White-Box-Testing
 - Graphenbasierte Testfallgenerierung
 - Schnittstellensignaturbasierte Testfallgenerierung
 - Testfallgenerierung nach logischen Kriterien
 - Syntaxbasierte Testfallgenerierung
- Testen eingebetteter Systeme

Lernergebnisse:

Die Studierenden kennen Qualitätsmerkmale und -kriterien und die Methoden zur Erreichung entsprechender Qualitätsziele. Sie können die gängigen Methoden und Verfahren zur Sicherstellung von Softwarequalität umsetzen.

- Kenntnis der wesentlichen Qualitätsmerkmale von Software und ihrer wechselseitigen Abhängigkeiten.
- Kenntnis der typischen Defizite der Software-Qualität und ihrer Gründe.
- Kenntnis der Aufgabenbereiche des Software-Qualitätsmanagement und Überblick über die wesentlichen Managementkonzepte.
- Kenntnis des Konzepts der Qualitätsmodelle und der relevanten Qualitätsmerkmale und -metriken.
- Überblick über mögliche Maßnahmen der Software-Qualitätssicherung, Kenntnis der wesentlichen konstruktiven und analytischen Maßnahmen der Software-Qualitätssicherung.
- Fähigkeit, ausgewählte Maßnahmen der Qualitätssicherung umzusetzen.
- Erkenntnis der besonderen Bedeutung der Usability als benutzerzentriertes Qualitätsmerkmal.
- Kenntnis der wesentlichen Ansätze, die Usability einer Software zu bewerten und zu gestalten.
- Kenntnis der methodischen Ansätze zur angemessenen Einbeziehung ergonomischer Aspekte in Software-Entwicklungsprozesse, insbesondere Potentiale und Probleme partizipativer Software-Entwicklung.
- Kenntnis von KI-Verfahren zur effektiven Qualitätssicherung von Software
- Erkenntnis der besonderen Bedeutung von Qualitätskriterien zum Testen von KI-generierten Anwendungen

Voraussetzungen und Empfehlungen:

Das Modul setzt grundlegende Kenntnisse der Softwareerstellung, insbesondere der Programmierung in einer höheren Programmiersprache, voraus.

Literatur:

- Ammann, Paul; Offutt, Jeff: Introduction to Software Testing. 1. Auflage. Cambridge, UK: Cambridge University Press, 2008.
- Balzert, Helmut: Lehrbuch der Softwaretechnik : Basiskonzepte und Requirements Engineering. 3. Auflage. Heidelberg: Spektrum Akademischer Verlag, 2009.
- Balzert, Helmut: Lehrbuch der Softwaretechnik : Softwaremanagement. 2. Auflage. Heidelberg: Spektrum Akademischer Verlag, 2008.
- Hoffmann, Dirk W.: Software-Qualität. Berlin: Springer-Verlag, 2008.
- Kneuper, Ralf: CMMI : Verbesserung von Software- und Systementwicklungsprozessen mit Capability Maturity Model Integration. Heidelberg: Dpunkt Verlag, 2007
- Kahn, Stephen H.: Metrics and Models in Software Quality Engineering. 2. Auflage. Boston (MA), USA: Addison-Wesley, 2002.
- Liggesmeyer, Peter: Software-Qualität : Testen, Analysieren und Verifizieren von Software. 2. Aufl. Heidelberg: Spektrum Akademischer Verlag, 2009.

- Schneider, Kurt: Abenteuer Softwarequalität : Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement. 2. Auflage. Heidelberg: Dpunkt Verlag, 2012
- Spillner, Andreas; Linz, Tilo: Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester Foundation Level nach ISTQB®-Standard. 7. Überarbeitete Auflage, Heidelberg, dpunkt Verlag, 2024
- Spillner, A.; Roßner, T; Winter, M; Linz, T.: Praxiswissen Softwaretest – Testmanagement: Aus- und Weiterbildung zum Certified Tester – Advanced Level nach ISTQB®-Standard. 4. Überarbeitete und erweiterte Auflage, dpunkt Verlag, Heidelberg, 2014
- Tian, Jeff : Software Quality Engineering. 1. Auflage. Hoboken (NJ), USA: John Wiley & Sons, 2005.
- Wallmüller, Ernest: Software Quality Engineering : Ein Leitfaden für bessere Software-Qualität. 3. Auflage. München: Carl Hanser Verlag, 2011.

Studiengänge:

- 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

◆ MB098 – Anwendungsentwicklung in ERP-Systemen

Verantwortliche:	Birger Wolter
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB037 – Anwendungsentwicklung in ERP-Systemen	Teilleistung:	TB038 – Ubg. Anwendungsentwicklung in ERP-Systemen
Lernform:	Vorlesung	Lernform:	Übung
Prüfungsform:	Klausur	Prüfungsform:	Abnahme
Prüfungsdauer:	120 Min.	Prüfungsdauer:	15 Min.
ECTS:	2,0	ECTS:	3,0
Benotung:	Drittelnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester	Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden	Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden	Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	0 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	45 Stunden	Flexibel einteilbarer Aufwand:	60 Stunden
Gesamtaufwand:	60 Stunden	Gesamtaufwand:	90 Stunden
Lehrende:	Emre Kilic	Lehrende:	Emre Kilic

Studieninhalte:

- Einführung und erste Schritte
- Werkzeuge der Entwicklungsumgebung
- Grundlegende Konzepte
- Datenverarbeitung
- Datenbankzugriffe
- Listenverarbeitung (Reports)
- Gestaltung von Bildelementen (Dynpros)
- Besonderheiten von Unternehmenssoftware
- Objektorientierte Programmierung mit ABAP Objects
- Webtechnologien: Web Dynpro, Fiori, Elements und UI5

Lernergebnisse:

Die Studierenden ...

- verstehen die spezifischen Anforderungen und Herausforderungen bei der Entwicklung von Unternehmenssoftware
- erlangen die Fähigkeit, SAP-Systeme zu verstehen und Erweiterungen für spezielle Anwendungsaufgaben vorzunehmen
- erwerben systematische Kenntnisse der ABAP-Programmiersprache, als bedeutender Träger der betriebswirtschaftlichen Anwendungslogik in SAP-Systemen
- können die grundlegenden Funktionen und Einsatzmöglichkeiten von SAP ABAP erklären sowie die Bedeutung der Programmiersprache innerhalb der SAP-Architektur verstehen
- beherrschen die verschiedenen Werkzeuge der ABAP-Entwicklungsumgebung und können diese effektiv bei der Entwicklung von Programmen einsetzen
- beherrschen die grundlegende Syntax und Struktur der ABAP-Programmiersprache sowie die verschiedenen Datentypen und Kontrollstrukturen
- können Datenbankabfragen und -manipulationen durchführen
- können Bildschirmmasken (Dynpros) entwerfen und implementieren, um Benutzereingaben zu erfassen und zu verarbeiten
- sind in der Lage objektorientierte Programme in ABAP zu entwickeln
- können Webanwendungen mit Web Dynpro und UI5 entwerfen und entwickeln
- verstehen die Prinzipien und die Bedeutung von Fiori und können Webanwendungen mithilfe von Fiori Elements erstellen

Die Studierenden können – durch Programmierübungen mit den Werkzeugen der Entwicklungsumgebung – die theoretisch erworbenen Kenntnisse praktisch umsetzen.

Voraussetzungen und Empfehlungen:

- **Grundlegende Programmierkenntnisse:** Vertrautheit mit einer oder mehreren Programmiersprachen, vorzugsweise mit Kenntnissen in einer objektorientierten Sprache wie Java oder C++.
- **Grundlagen der Softwareentwicklung:** Verständnis der grundlegenden Konzepte und Prinzipien der Softwareentwicklung, einschließlich Algorithmen, Datenstrukturen und Softwareentwicklungsprozesse.
- **Datenbankkenntnisse:** Basiswissen über relationale Datenbanken und SQL, da die ABAP-Programmierung häufig Datenbankzugriffe beinhaltet.
- **SAP-Grundkenntnisse:** Grundlegendes Verständnis der SAP-Systemlandschaft und -Module ist von Vorteil.

Literatur:

- BANDARI, Kiran; Complete ABAP. Rheinwerk Verlag, 3. Auflage 2023
- ENGBRECHT, Michael; SAP Fiori: Implementierung und Entwicklung - User Experience, Design Thinking, SAP Gateway. Rheinwerk Verlag, 3. Auflage 2020
- ROTH, Felix; ABAP - Das umfassende Handbuch: Konzepte, Sprachelemente und Werkzeuge in ABAP. Rheinwerk Verlag, 3. Auflage 2023
- KELLER, Horst; KRÜGER, Sascha; ABAP Objects. ABAP-Programmierung mit SAP NetWeaver. Rheinwerk Verlag vorm. Galileo Press, 3. Auflage 2006
- KELLER, Horst; THÜMMEL, Wolf Hagen; ABAP - Das umfassende Handbuch. Rheinwerk Verlag vorm. Galileo Press, 1. Auflage 2014

Studiengänge:

- Informatik Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB118 – Soft Skills

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

Bestandteile:

Teilleistung:	TB042 – Assistenz	Teilleistung:	TB043 – Communication Skills
Lernform:	Assistenz	Lernform:	Workshop
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)	Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	2,0	Prüfungsdauer:	2,0
ECTS:	2,0	ECTS:	2,0
Benotung:	Bestanden/nicht Bestanden	Benotung:	Bestanden/nicht Bestanden
Turnus:	jedes Semester	Turnus:	jedes Semester
Dauer (pro Termin):	3 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	22 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	68 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden	Gesamtaufwand:	60 Stunden
Lehrende:	Jan-Paul Lüttke	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 selbständig 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 Assistenzfähigkeit.

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

◆ MB123 – Prozessmodellimplementation

Verantwortliche:	Christian Uhlig
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB049 – Prozessmodellimplementation
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Abnahme
Prüfungsdauer:	60 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:	0 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	135 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Christian Uhlig

Studieninhalte:

- Architekturen betrieblicher Anwendungssysteme
 - Client-Server-Architekturen mit persistenter Datenhaltung
 - Web-Applikationen und Web-Services
- Objektorientierte Web-Anwendungsentwicklung mit Java
 - Objekt-relationale Abbildung mit der Java Persistence API
 - Architekturen und Frameworks für Web-Anwendungen
- Implementierung von Geschäftsprozessen
 - ausführbare BPMN-Modelle
 - Automatisierung von Geschäftsprozessen per SOA
 - Web-Services zur Verwaltung von Geschäftsobjekten ("REST-Services")
- Modelltypen der ARIS-Methode zur Verbindung von Software-Engineering mit Geschäftsprozessmodellierung
- Praktische Aufgabenstellungen
 - gegebene Modelle zu einem Fachkonzept und zur Spezifikation eines zu implementierenden Anwendungssystems (Prozess- und Datenmodellierung, Maskendesign, Klassendiagramm, usw.)
 - Entwurf und Implementierung eines Anwendungssystems zu einem Ausschnitt des modellierten Fachkonzeptes (Datenhaltung, Backend, Webanwendung)
 - Automatisierung eines per Modell vorgegebenen Geschäftsprozesses mittels BPMN

Lernergebnisse:

Die Studierenden ...

- skizzieren die Grundlagen von Softwareprojekten und des Softwareentwicklungszyklus (Spezifikation, Entwurf und Implementierung).
- setzen Geschäftsprozessmodelle (EPK, BPMN) zur Fachkonzeptbeschreibung in Beziehung zum Entwurf und zur Implementierung unterstützender Anwendungssysteme.
- verbinden Geschäftsprozessmodelle (BPMN) mit Modell-Elementen des Softwareentwurfs (z. B. Datenobjekte, Maskendesign, Klassenhierarchie, usw.).
- identifizieren und entwerfen die notwendigen Bedienelemente (insbesondere Bildschirmmasken) zur Unterstützung bestimmter Funktionen eines gegebenen Geschäftsprozesses.
- differenzieren Desktop- und Web-Anwendungen und ihre wesentlichen Merkmale.
- vergleichen kritisch verschiedene Ansätze zur Persistierung und zur Umsetzung von Web-Anwendungen.
- wählen aus und nutzen Konzepte, Programmierschnittstellen und Frameworks der Java-Landschaft zur Erstellung webbasierter Anwendungssysteme mit Datenbankunterstützung (J2EE, JPA, JTA, usw.).
- entwerfen und entwickeln datenbankgestützte Web-Anwendungen zur Unterstützung von Geschäftsprozessen auf Basis der Programmiersprache Java und wichtiger damit verbundener Konzepte und Frameworks.
- setzen eine Modelllandschaft in Bezug zu Einheiten der Softwaretechnik (Klassen zu Business Objects, Klassen zu Masken, usw.) und leiten dabei wesentliche Teile des Systemgerüsts systematisch aus Modellinhalten ab.
- modellieren ausführbare BPMN-Modelle zu Geschäftsprozessen und bringen sie unter Einsatz eines Workflow Management Systems zur Ausführung.
- implementieren einfache Web Services als Bausteine ausführbarer Prozessmodelle.

Voraussetzungen und Empfehlungen:

Es wird vorausgesetzt, dass die Teilnehmer Modelle zur Repräsentation verschiedener Sichten auf Geschäftsprozesse (BPMN Process Diagram, BPMN Collaboration Diagram, ERM/ERD) interpretieren können. Weiterhin wird vorausgesetzt, dass die Teilnehmer die Programmiersprache Java beherrschen und praktische Kenntnisse zu relationalen Datenbanken (insbesondere Formulierung von SQL-Statements und Erstellung von Datenbank-Schemata) und zur Erstellung von Web-Anwendungen besitzen.

Literatur:

- Krüger, Jörg; Uhlig, Christian: Praxis der Geschäftsprozessmodellierung - ARIS erfolgreich anwenden, VDE Verlag, 2009
- Lehmann, Frank: Integrierte Prozessmodellierung mit ARIS, dpunkt.verlag, 2007
- Scheer, August-Wilhelm: ARIS Vom Geschäftsprozess zum Anwendungssystem, 4. Auflage, Springer, 2002
- Scheer, August-Wilhelm: ARIS Modellierungsmethoden, Metamodelle, Anwendungen, 4. Auflage, Springer, 2001
- Müller, Bernd; Wehr, Harald: Java Persistence API 2: Hibernate, EclipseLink, OpenJPA und Erweiterungen, Carl Hanser Verlag, 2012
- DeMichiel, Linda; Shannon, Bill: Java Platform, Enterprise Edition (Java EE) Specification v7, Oracle 2013
- Jakarta Persistence Team: Jakarta Persistence API, Version 3.1, 2022
- OMG: Business Process Model and Notation (BPMN) Version 2.0, 2011
- Software AG: ARIS-Dokumentation (Methodenhandbuch, Bedienhandbücher), jeweils aktuellste Fassung
- jeweils aktuelle Dokumentation zum Spring Framework
- jeweils aktuelle Dokumentation zum Vaadin Framework
- jeweils aktuelle Dokumentation zum Camunda BPM Workflow Management System

Studiengänge:

- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- IT-Management / -Consulting & -Auditing Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB244 – Exploratory Data Analysis

Verantwortliche:	Hendrik Annuth
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB079 – Data Visualization, Feature Engineering	Teilleistung:	TB090 – Ubg. Exploratory Data Analysis
Lernform:	Vorlesung	Lernform:	Übung
Prüfungsform:	Klausur	Prüfungsform:	Abnahme
Prüfungsdauer:	90 Min.	Prüfungsdauer:	30 Min.
ECTS:	2,0	ECTS:	2,0
Benotung:	Drittelnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester	Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	75 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	45 Stunden
Aufwand während Semesterferien:	0 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	90 Stunden	Gesamtaufwand:	60 Stunden
Lehrende:	Hendrik Annuth	Lehrende:	Marco Pawlowski

Studieninhalte:

- Machine-Learning-Verfahren Naive Bayes und Random Forest
- Data Transformation
- Data Augmentation
- Feature Embedding
- Feature Analysis: Feature Importance, Partial Dependencies; Data Leakage
- Unstructured Data
- Zeitreihen
- Erweiterte Einführung in R und Python
- Wichtigste Pakete zur Datenverarbeitung in R und Python
- Verwendung der gängigsten Visualisierungsverfahren in R und Python
- Erstellung von Diagrammen, Graphen und Histogrammen
- Zusammenstellung von Informationsgrafiken zur gleichzeitigen Darstellung
 - Das Pyramidenprinzip
 - Auswahlkriterien für Balken-, Torten- und Streudiagramme, Histogramme, Graphen
 - Metriken und Dashboards
 - Einführung in die Datenreinigung, -aufbereitung und -erweiterung
 - Einführung in die Topologie
 - Automatisierte Visualisierungsanwendungen
 - Clustering Methoden k-Means, Self-organizing Maps, Hierarchical Clustering
 - Dimensionalitätsreduktionsverfahren: PCA, T-SNE, UMAP

Lernergebnisse:

- Ein gutes Verständnis, wie beliebige Daten in einen Vektor umgewandelt bzw. als solches aufgefasst werden können, um verschiedene Datenarten für den Einsatz von Machine-Learning-Verfahren nutzbar zu machen
- Sicheres Beherrschen von Verfahren zur Datenreinigung, Data Augmentation und Datentransformation, um bewusst Problemen im Lernprozess vorzubeugen
- Kenntnis von Normalisierungstechniken, um die Ergebnisse von Machine-Learning-Verfahren zu verbessern
- Ein solides Verständnis von kategorischen und kontinuierlichen Datentypen, deren Konvertierung, von Einstellungsparametern und deren Auswirkungen auf ein Machine-Learning-Verfahren
- Ein Verständnis und Lösungsansätze beim Umgang mit Daten, die Zeitreihen enthalten
- Einsatz von Visualisierungstechniken, um aus Datensätzen verbesserten inhaltlichen Nutzen ziehen zu können
- Bewusstsein für die gezielte Auswahl von Graphen, Diagrammen und Tabellen zum Zwecke der Hervorhebung von Erkenntnissen aus Datensätzen
- Grundverständnis für die Funktionsweise aktueller Softwarelösungen und Pakete für die Visualisierung und Aufbereitung von Daten
- Erschließung von Datensätzen und deren Besonderheiten durch die Programmiersprachen R und Python
- Die Fähigkeit, Visualisierungstechniken im Kontext der Datensatzanalyse so einzusetzen, dass Datensätze durch den Erkenntnisgewinn sinnvoll aufbereitet und erweitert werden können
- Verständnis zur Visualisierung hochdimensionaler Daten
- Sicherer Umgang mit der Programmierung in R und Python und den dazugehörigen Entwicklungsumgebungen
- Erfahrung mit dem Laden, Verarbeiten und Visualisieren von komplexen Datensätzen
- Fähigkeit zur programmatischen Erstellung von verschiedenen Diagrammen, Graphen und Histogrammen
- Erfahrung mit dem Zusammenstellen verschiedener Informationsgrafiken im Kontext der Aufbereitung einer zielgerichteten Fragestellung

Voraussetzungen und Empfehlungen:

Das Modul setzt erweiterte Programmierkenntnisse voraus. Sprachelemente aus Python und R werden vorgestellt und verwendet. Dabei werden Konzepte wie Zeiger, Speicherreservierung und Garbage-Collection als bekannt vorausgesetzt. Mathematisches Grundwissen aus den Veranstaltungen Mathematik 1 und 2, aus den Gebieten Statistik und Linearer Algebra wird in der Veranstaltung vorausgesetzt.

Literatur:

- Feature Engineering and Selection: A Practical Approach for Predictive Models; Chapman and Hall 2019; Max Kuhn and Kjell Johnson;

- Feature Engineering for Machine Learning Models: Principles and Techniques for Data Scientists; O'Reilly 2018; Alice Zheng, Amanda Casari
- The Art of Feature Engineering Essentials for Machine Learning; Cambridge University Press 2020; Pablo Duboue

Siehe auch Veranstaltung Machine Learning

- Visual Display of Quantitative Information; Bertrams 2001; Edward R Tufte
- Say It With Charts: The Executives's Guide to Visual Communication: The Executive's Guide to Visual Communication; McGraw-Hill ; Gene Zelazny
- The Pyramid Principle: Logic in Writing and Thinking: Logical Writing, Thinking and Problem Solving; Financial Times Series 1996; Barbara Minto
- The Elements of Statistical Learning; Springer 2009; Trevor Hastie, Robert Tibshirani, Jerome Friedman
- Datenvisualisierung: Vom Diagramm zur Virtual Reality; UTB 2018; Peter Fischer-Stabel
- Storytelling mit Daten: Die Grundlagen der effektiven Kommunikation und Visualisierung mit Daten; Vahlen 2017; Cole Nussbaumer Knaflitz, Mike Kauschke
- The Truthful Art: Data, Charts, and Maps for Communication (Voices That Matter); New Riders 2016; Alberto Cairo
- Infografik: Komplexe Daten professionell visualisieren; Rheinwerk Design 2018; Raimar Heber
- Data Visualization: A Practical Introduction; Princeton University Press 2019; Kieran Healy
- Datenvisualisierung mit Tableau; mitp 2018; Alexander Loth
- Learning Python; O'Reilly and Associates 2013; Mark Lutz
- Basic Elements of Computational Statistics; Springer 2017; Wolfgang Karl Härdle, Ostap Okhrin, Yarema Okhrin
- Data Science mit Python: Das Handbuch für den Einsatz von IPython, Jupyter, NumPy, Pandas, Matplotlib und Scikit-Learn; mitp 2017; Jake VanderPlas

Studiengänge:

- Angewandte Wirtschaftspsychologie & Data Analytics Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- IT-Management & Consulting Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science
- Wirtschaftsinformatik Bachelor of Science

◆ MB268 – Projekt App- und Web-Development

Verantwortliche:	Christian-Arved Bohn
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB259 – Projekt App- und Web-Development
Lernform:	Projektarbeit
Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	30 Min.
ECTS:	3,0
Benotung:	Drittelnoten
Turnus:	jedes Semester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	135 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Christian-Arved Bohn

Studieninhalte:

Das Projekt führt in aktuelle Prinzipien der Programmierung für mobile Devices und Web-Browser ein. Anhand einer konkreten Aufgabenstellung wird eine komplette Applikation entwickelt, in der aktuelle Entwicklungsumgebungen und Programmpakete analysiert und auf die konkrete Anwendung bezogen selektiert werden. Deren Verknüpfung zu einer Applikation im Rahmen eines Client-Server-Modells ist zentraler Fokus des Projekts. Hierbei wird besonderer Wert auf Aktualität und einer effizienten Kombination der avisierten Bibliotheken gelegt.

Lernergebnisse:

Studenten und Studentinnen

- sind in der Lage, die in Ihrem Studium gewonnene Programmiererfahrung im Rahmen eines Client-Server Modells zu verwirklichen, das effizient und robust auf mobilen Devices ausführbar ist,
- beherrschen den sicheren Umgang mit hochaktuellen Bibliotheken der Web-Programmierung,
- besitzen durch das Arbeiten an separaten Programmmodulen innerhalb eine mittelgroßen Gruppe ein gesteigerte Teamfähigkeit.

Voraussetzungen und Empfehlungen:

Erfahrung in Programmierung, Vorlesung "Programmstrukturen 1".

Literatur:

- Aichele, Schönberger: App-Entwicklung – effizient und erfolgreich: Eine kompakte Darstellung von Konzepten, Methoden und Werkzeugen, Springer 2016
- Ben Frain: Responsive Web Design with HTML5 and CSS: Build future-proof responsive websites using the latest HTML5 and CSS techniques, Packt Publishing, 2022
- HTML und CSS: Das umfassende Handbuch zum Lernen und Nachschlagen. Inkl. JavaScript, Responsive Webdesign, React und Angular u. v. m., Rheinwerk Computing, 2023

Studiengänge:

- Informatik Bachelor of Science
- Medieninformatik Bachelor of Science

◆ MB293 – Digital Product Management

Verantwortliche:	Atilla Wohllebe
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB284 – Digital Product Management
Lernform:	Vorlesung
Prüfungsform:	Klausur + ggf. Bonus
Prüfungsdauer:	60 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 einstellbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Atilla Wohllebe

Studieninhalte:

- Einführung Digital Product Management
- Konzeption von Produkten
 - Erarbeitung einer Product Vision
 - Lösungsentwicklung im Rahmen der Product Discovery (Design Sprint)
 - Validierung von Produktideen am Markt
- Datengetriebene Produktentwicklung
 - Entwicklung und Weiterentwicklung von Produkten
 - Agile Arbeitsweisen (Scrum, Kanban)
 - Erhebung und Priorisierung von Anforderungen
 - DevOps - Grundlegende Gedanken und Methoden

Lernergebnisse:

Die Studierenden...

- verfügen über weitgehende Kenntnisse zum Management digitaler Produkte und Services, insbesondere mit Blick auf deren konzeptionelle (Weiter-) Entwicklung.
- sind in der Lage, mit Hilfe geeigneter Methoden eine Product Vision zu entwickeln und diese im Rahmen einer Product Discovery in konkrete Lösungsansätze zu übersetzen und zu validieren.
- verfügen außerdem über Kenntnisse agiler Arbeitsweisen und aktueller Vorgehensweisen in der Softwareentwicklung und können diese in den Kontext des Digital Product Managements einordnen.

Voraussetzungen und Empfehlungen:

Keine

Literatur:

- Düsterbeck, F., Einemann, I. (2022). Product Ownership meistern - Produkte erfolgreich entwickeln. dpunkt.verlag, Heidelberg.
- Hoffmann, S. (2020). Digitales Produktmanagement: Methoden - Instrumente - Praxisbeispiele. Springer Gabler, Wiesbaden. <https://link.springer.com/book/10.1007/978-3-658-30629-8>
- Kim, G., Humble, J., Debois, P., Willis, J., Forsgren, N. (2022). Das DevOps Handbuch - Teams, Tools und Infrastrukturen erfolgreich umgestalten. O'Reilly / dpunkt.verlag, Heidelberg.
- Kittlaus, H., Clough, P. (2009). Software Product Management and Pricing - Key Success Factors for Software Organizations. Springer Berlin, Heidelberg. <https://link.springer.com/book/10.1007/978-3-540-76987-3>
- Pranam, A. (2018). Product Management Essentials - Tools and Techniques for Becoming an Effective Technical Product Manager. Apress Berkeley, CA. <https://link.springer.com/book/10.1007/978-1-4842-3303-0>
- Wagenblatt, A. (2019). Software Product Management - Finding the Right Balance for YourProduct Inc.. Springer Cham. <https://link.springer.com/book/10.1007/978-3-030-19871-8>

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-Management & Consulting Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Medieninformatik Bachelor of Science
- Smart Technology Bachelor of Science

◆ MB313 – Computer Vision

Verantwortliche:	Dennis Säring
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB305 – Computer Vision
Lernform:	Vorlesung mit integrierter Übung
Prüfungsform:	Klausur
Prüfungsdauer:	60 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:	125 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	155 Stunden
Lehrende:	Dennis Säring Marco Pawlowski

Studieninhalte:

- Image fundamentals and processing (Convolution, Fourier, Resampling)
- Image classification with machine learning (Semantic gap, K-NN)
- Convolutional neural networks (CNNs)
- CNN architectures (AlexNet, VGG, GoogLeNet, ResNet)
- Recurrent Neural Networks (RNNs)
- Vision Transformer (ViTs)
- Visualization and Understanding (Explainable AI)
- Object detection (R-CNNs, YOLO)
- Semantic and Instance segmentation (U-Net, Mask R-CNN)
- Generative models (Autoencoder, GANs)

Lernergebnisse:

- Grundlegende Kenntnisse über den Einsatzbereich und die Möglichkeiten von Computer Vision in der Bildverarbeitung (am Beispiel der Medizinischen Bildverarbeitung)
- Kenntnisse klassischer Methoden der Bildverarbeitung u.a. Faltung und Transformation von Bildern
- Kenntnisse etablierter Methoden zur Objektdetektion und Klassifikation von Objekten in Bildern und Bewegtbildern
- Überblick über aktuelle Entwicklungstrends in Computer Vision mit Fokus auf Deep-Learning
- Kenntnisse ausgewählter CNN Architekturen, deren Bestandteile sowie deren Anwendungsbereiche im Bereich von Computer Vision
- Kenntnis der aktuellen Softwaretools und Bibliotheken zur Implementierung von CNNs
- Fähigkeit, die zuvor erlernten theoretischen Kenntnisse an einem Beispielprojekt praktisch umzusetzen
- Fähigkeit, die berechneten Modelle zu evaluieren und deren Qualität kritisch beurteilen zu können.

Voraussetzungen und Empfehlungen:

Das Modul baut auf der Veranstaltung "Machine Learning" auf und setzt insbesondere das Verständnis vom Gradientenabstiegsverfahren, von Optimierungstechniken im Lernprozess, von Normalisierungstechniken und von neuronaler Netzwerkarchitektur voraus.

Literatur:

- Bildverarbeitung und Objekterkennung: Computer Vision in Industrie und Medizin von: Herbert Süße, Erik Rodner
- Modern Deep Learning and Advanced Computer Vision: A Perspective Approach von: Dr. P. S. Jagadeesh Kumar, Prof. Thomas Binford
- Fundamentals of Deep Learning and Computer Vision: A Complete Guide to become an Expert in Deep Learning and Computer Vision von: Nikhil Singh, Paras Ahuja

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science

◆ MB379 – Konzepte der funktionalen Programmierung

Verantwortliche:	Torben Tietgen
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB375 – Konzepte der funktionalen Programmierung	Teilleistung:	TB376 – Übg. Konzepte der funktionalen Programmierung
Lernform:	Vorlesung	Lernform:	Übung
Prüfungsform:	Klausur	Prüfungsform:	Abnahme
Prüfungsdauer:	90 Min.	Prüfungsdauer:	10 Min.
ECTS:	2,0	ECTS:	3,0
Benotung:	Drittelnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Wintersemester	Turnus:	Wintersemester
Dauer (pro Termin):	2 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	25 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	78 Stunden
Aufwand während Semesterferien:	20 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden	Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	60 Stunden	Gesamtaufwand:	93 Stunden
Lehrende:	Torben Tietgen	Lehrende:	Malte Heims

Studieninhalte:

- Typen
 - Typklassen
 - Funktionsdefinitionen
 - Rekursive Funktionen
 - Funktionen höherer Ordnung
 - Korrektheitsargumentationen
 - Selbstdefinierte Datenstrukturen und Bäume
- Zusätzlich werden in der Übung praxisrelevante Aspekte der Anwendungsentwicklung mit der Programmiersprache Haskell behandelt, die nicht Bestandteil der Vorlesung sind. Die Bearbeitung der Übungsaufgaben folgt parallel zum Stoff der Vorlesung in Zweiergruppen mit Abnahme und Diskussion der Lösungen.

Lernergebnisse:

- Die Studierenden ...
- kennen die Programmierparadigmen der funktionalen Programmierung und der Algorithmenkonstruktion ohne Programmvariablen, Zuweisungen und Schleifen.
 - besitzen das Verständnis und den Umgang mit Rekursion beim Entwickeln von Algorithmen.
 - kennen die Vorteile einer streng getypten Programmiersprache bei der Programmentwicklung und beim Erlernen einer neuen Programmiersprache.
 - haben das erste Grundverständnis über das Führen von Korrektheitsargumentationen wie Terminierung und Ableitung von Gesetzen und Programmeigenschaften erworben.
 - sind geschult bezüglich der Fähigkeiten zur Abstraktion und zur Modellbildung.
 - haben das sorgfältige, methodische und systematische Programmieren im Kleinen trainiert.

Voraussetzungen und Empfehlungen:

Grundlegende Programmiererfahrungen in einer beliebigen Programmiersprache sind empfehlenswert, aber nicht vorausgesetzt. Zudem ist die Installation von Haskell auf dem eigenen Rechner empfehlenswert.

Literatur:

- Miran Lipovaca: Learn You a Haskell for Great Good!: A Beginner's Guide, No Starch Press, 2011, ISBN: 978-1593272838
- Graham Hutton: Programming in Haskell, Cambridge University Press, 2016, ISBN: 978-1316626221

Studiengänge:

- Data Science & Artificial Intelligence Bachelor of Science
- Informatik Bachelor of Science
- Smart Technology Bachelor of Science

◆ MB053 – Datenschutz und Medienrecht

Verantwortliche:	Gerd Beuster
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB078 – Datenschutz, Medienrecht
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	180 Min.
ECTS:	5,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	135 Stunden
Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Jens Brelle Behrang Raji

Studieninhalte:

Medienrecht

- Einführung und Rechtsgrundlagen
- Medienfreiheit und Individualrechte
- Grundlagen des Medien- und Äußerungsrechts
- Grundlagen des Urheberrechts
- Überblick Gewerbliche Schutzrechte (Marken, Designs, Patente)
- Grundlagen des Werbe- und Wettbewerbsrechts
- Rechtsverletzungen und Folgen
- Onlinerecht: Domainrecht, Internetrecht, Social Media-Recht, KI-Recht
- Grundlagen des Lizenzvertragsrechts
- Grundlagen Software- und Projektverträge
- Medienrechtliche Praxis: Aktuelle Entwicklungen und wichtige Urteile

Datenschutz

- Gesetzliche Grundlagen des Datenschutzes
 - Anwendung und praktische Umsetzung des Bundesdatenschutzgesetzes (BDSG)
 - Wesentliche Grundlagen aus ausgewählten bereichsspezifischen und bereichsübergreifenden Datenschutzgesetzen
 - Rechte, Pflichten und Aufgaben des betrieblichen Datenschutzbeauftragten zur Einrichtung des Datenschutzmanagements
 - Datenschutz in der Werbepraxis
- Technisch-organisatorischer Datenschutz
 - Grundanforderungen und Grundfunktionen der IT-Sicherheit in Bezug auf die Anforderungen der Datenschutzgesetze
 - Risikomanagement und Schlüsseltechnologien zur Realisierung des technisch-organisatorischen Datenschutzes
 - Kosten-/Nutzen des Datenschutzes
 - Verfahren zur Umsetzung des gesetzlichen Anforderungen des technisch-organisatorischen Datenschutzes
 - Auswahlverfahren zu geeigneten und angemessenen IT-Sicherheitsmechanismen

Lernergebnisse:

Die Studierenden erarbeiten im Rahmen der Lehrveranstaltung Verständnis für die Grundzüge des Medien- und des Urheberrechts sowie für angrenzende Rechtsgebiete des gewerblichen Rechtsschutzes. Sie verstehen zwischen den unterschiedlichen Rechtsgebieten zu unterscheiden und erwerben Kenntnisse praxisnaher Anwendungsfelder, die ein nachhaltiges Problembewusstsein schaffen, auf dessen Grundlage juristische Fragestellungen identifiziert und auf Rechtsanwendungsebene beurteilt werden können.

Die Studierenden sind fähig, in ihrem späteren Wirkungskreis datenschutzrechtliche Fragestellungen einzuordnen, um bei Bedarf auf Spezialistenunterstützung gezielt zurückgreifen zu können. Erwerb eines Grundlagenwissens im nationalen und europäischen Datenschutzrecht.

Voraussetzungen und Empfehlungen:

Lesekompetenz wird vorausgesetzt. Fachkompetenz im Gebiet Recht ist nicht notwendig.

Literatur:

Recht allgemein

- Engisch, Einführung in das juristische Denken, 12. Aufl. 2018 (Bearbeiter: Würtenberger/Otto).

Kommentare Datenschutzrecht

- Freund/Schmidt/Hepp/Roschek, DSGVO Praxis-Kommentar, 1. Aufl. 2022
- Kühling/Buchner, DSGVO
- Taeger/Gabel, DSGVO – BDSG – TDDDG
- Simitis/Hornung/Spiecker gen. Döhmman, DSGVO mit BDSG

Einführungen Datenschutzrecht

- BfDI, Info 1, 2020 – Gesetzestexte und Erläuterungen (<https://www.bfdi.bund.de/SharedDocs/Publikationen/Infobroschueren/INFO1.pdf>)

Medienrecht

- Fechner / Mayer
Vorschriftensammlung zum Medienrecht
19. Auflage 2025, Verlag C., F. Müller
- Prof. Dr. Axel Beater (Uni Greifswald)
Skript Medienrecht (2024/2025)
https://rsf.uni-greifswald.de/storages/uni-greifswald/fakultaet/rsf/lehrstuehle/lb-beater/MedienR_WS_24-25.pdf
- Prof. Dr. Hubertus Gersdorf (Uni Leipzig)
Skript Medienrecht (2021/2022)
https://www.jura.uni-leipzig.de/fileadmin/Fakult%C3%A4t_Juristen/Professuren/Gersdorf/Dokumente/Lehrveranstaltungen/Vorlesungen/Medienrecht_I/WiSe_2021_2022/Medienrecht_I_WiSe_2021_2022.pdf
- Prof. Dr. Thomas Hoeren
Skriptum Internetrecht (Stand März 2023)
Skriptum IT-Vertragsrecht (zusammen mit Stefan Pinelli) Mai 2024
<https://www.itm.nrw/lehre/materialien/>
- Prof. Dr. Ansgar Ohly (LMU München)
Skripten zu den Immaterialgüterrechten:
 - Einführung in das Recht des geistigen Eigentums
 - Urheberrecht
 - Markenrecht
 - Lauterkeitsrecht
 - Patentrecht
 International Intellectual Property Law
<https://www.jura.lmu.de/de/fakultaet/lehrstuehle/lehrstuehle-und-professuren-fuer-buergerliches-recht/lehrstuhl-prof.-ohly>

Studiengänge:

- Computer Games Technology Bachelor of Science
- Data Science & Artificial Intelligence Bachelor of Science
- E-Commerce Bachelor of Science
- Informatik Bachelor of Science
- Marketing, Media & AI Bachelor of Science
- Medieninformatik Bachelor of Science

◆ MB058 – Software-Design

Verantwortliche:	Christian Uhlig
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB026 – Software-Design
Lernform:	Vorlesung
Prüfungsform:	Klausur
Prüfungsdauer:	90 Min.
ECTS:	2,0
Benotung:	Drittelnoten
Turnus:	Sommersemester
Dauer (pro Termin):	4 Semesterwochenstunden
Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	30 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	60 Stunden
Aufwand während Semesterferien:	60 Stunden
Flexibel einstellbarer Aufwand:	0 Stunden
Gesamtaufwand:	150 Stunden
Lehrende:	Christian Uhlig

Studieninhalte:

- Einordnung und Bedeutung des Entwurfs im Softwareentwicklungsprozess
- Informelle Modellierungstechniken im Software-Entwurf: OMT, UML, ERM/ERD
- Verträge zwischen Softwarekomponenten, Design by Contract
- Objektorientierte Entwurfsmuster
 - Erzeugungsmuster
 - Strukturmuster
 - Verhaltensmuster
- Fallstudien
- Serviceorientierte Architektur

Lernergebnisse:

Die Studierenden ...

- erkennen und erläutern die Einordnung des Entwurfs in den Softwareentwicklungsprozess.
- erkennen und erläutern die Bedeutung der Modellbildung im Softwaredesign.
- erkennen und erläutern die Bedeutung von Verträgen bei Entwurf und Implementierung abstrakter Datentypen.
- erkennen und erläutern das Entwurfparadigma Design by Contract am Beispiel der Programmiersprachen Eiffel und Java.
- differenzieren Vererbung und Komposition als zentrale Konzepte des objektorientierten Entwurfs und wählen problemadäquat aus.
- wenden informelle Notationen und Methoden (OMT, UML, ERD, ...) zur Modellierung eines Softwaresystems an.
- wenden formale Notationen (z.B. Haskell) zur Definition der Datenstrukturen und der Schnittstellen eines Softwaresystems an.
- differenzieren Entwurfsmuster auf Grundlage von Struktur, Motivation und Zielsetzung.
- wenden typische objektorientierte Entwurfsmuster zur Lösung von softwaretechnischen Problemstellungen an.
- nennen und erläutern sprachabhängige und -unabhängige Implementierungsaspekte bei der Anwendung von Entwurfsmustern.
- erkennen und erläutern die Anwendbarkeit und Kombinierbarkeit einzelner Entwurfsmuster.
- differenzieren Flexibilität und Effizienz bei der problembezogenen Auswahl und Anwendung von Entwurfsmustern.
- nennen und erläutern die Grundzüge der serviceorientierten Architektur.

Voraussetzungen und Empfehlungen:

Vorausgesetzt werden Grundkenntnisse in imperativen und speziell objektorientierten Programmiersprachen sowie konkret in der Programmiersprache Java. Dies umfasst insbesondere objektorientierte Sprachelemente (Schnittstellen, abstrakte Klassen, konkrete Klassen) und Konzepte der Wiederverwendung (insbesondere Vererbung und dynamisches Binden).

Literatur:

- Balzert, Helmut: Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, 3. Auflage, Spektrum, 2011
- Balzert, Helmut; Balzert, Heide; Koschke, Rainer; Lämmel, Uwe; Liggesmeyer, Peter; Quante, Jochen: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, 3. Auflage, Spektrum, 2009
- Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John: Design Patterns: Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software, mitp, 2014
- Freeman, Eric; Robson, Elisabeth; Bates, Bert; Sierra, Kathy: Head First Design Patterns, O'Reilly, 2014
- Rumbaugh, James; Blaha, Michael; Premerlani, William; Eddy, Frederick; Lorensen, William: Objektorientiertes Modellieren und Entwerfen, Hanser, 1994
- Fowler, Martin: UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3. revidierte Auflage, Addison-Wesley, 2003

Studiengänge:

- Computer Games Technology 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

◆ MB101 – Echtzeitsysteme

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	TB063 – Echtzeitsysteme, Interface-Technologie	Teilleistung:	TB070 – Prakt. Echtzeitsysteme
Lernform:	Vorlesung	Lernform:	Praktikum
Prüfungsform:	Klausur	Prüfungsform:	Abnahme
Prüfungsdauer:	150 Min.	Prüfungsdauer:	30 Min.
ECTS:	3,0	ECTS:	2,0
Benotung:	Drittnoten	Benotung:	Bestanden/nicht Bestanden
Turnus:	Sommersemester	Turnus:	Sommersemester
Dauer (pro Termin):	2 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	4 Termine
Zeit in Veranstaltungen:	15 Stunden	Zeit in Veranstaltungen:	5 Stunden
Sonstiger Arbeitsaufwand während Vorlesungszeit:	65 Stunden	Sonstiger Arbeitsaufwand während Vorlesungszeit:	5 Stunden
Aufwand während Semesterferien:	5 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	5 Stunden	Flexibel einteilbarer Aufwand:	50 Stunden
Gesamtaufwand:	90 Stunden	Gesamtaufwand:	60 Stunden
Lehrende:	Dennis Siring Sergei Sawitzki	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 Programmmentwurf
 - 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:

- Böresök: 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

◆ MB121 – Software-Projekt

Verantwortliche:	Ulrich Hoffmann
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:	TB047 – Softwareprojekt
Lernform:	Projektarbeit
Prüfungsform:	Praktikumsbericht / Protokoll
Prüfungsdauer:	20 Min.
ECTS:	8,0
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:	0 Stunden
Flexibel einteilbarer Aufwand:	210 Stunden
Gesamtaufwand:	240 Stunden
Lehrende:	Christian-Arved Bohn

Studieninhalte:

Im Rahmen der digitalen Transformation werden wiederkehrende Aufgaben zunehmend automatisiert. Einmalige Tätigkeiten hingegen lassen sich schwierig automatisieren und werden daher in Zukunft weiter an Bedeutung gewinnen. Diese einmaligen, temporären Aufgaben sind per Definition Projekte; nicht zuletzt aufgrund dieses Umstandes wird Projektmanagement eine der wichtigsten Fähigkeiten für eine erfolgreiche Karriere im 21. Jahrhundert.

Wie schwierig Projektmanagement in der Praxis ist, wird beispielsweise dadurch ersichtlich, dass mehr als 2 von 3 IT-Projekten ihre Ziele verfehlen. Auch wenn die Gründe hierfür im Einzelfall sehr unterschiedlich sein mögen, lassen diese sich doch in zwei breite Gruppen unterteilen. Zur ersten Gruppe zählen fehlende Projektmanagement-Kompetenzen wie Auswahl der Projektmethodik, Projektplanung oder Risikokontrolle. Zur zweiten Gruppe zählen ungenügende Soft Skills, um alle beteiligten Stakeholder zu managen, wie beispielsweise Motivation, Konfliktlösung oder Veränderungsmanagement.

Zielsetzung dieser Veranstaltung ist die Entwicklung wesentlicher Grundlagen in beiden Bereichen: Grundlegende Projektmanagementfähigkeiten (im engeren Sinne) sowie notwendige Softskills eines Projektleiters.

Kurzzugliederung:

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

Variierende Themen für Software-Projekte, unter anderem aus dem Web-Bereich, die mit objektorientierten Techniken zu lösen sind.

Lernergebnisse:

Ziel dieses Projekts ist es, den Einsatz von von OO-Techniken in einer größeren Aufgabe in einem Team zu üben. Dabei ist die Themenstellung so gewählt dass zwischen den verschiedenen 2-er-Gruppen die Notwendigkeit der Koordination, Abstimmung und Diskussion, insbesondere über die Schnittstellen der Teilaufgaben, besteht.

Weiter sollen die Modellierung und der Entwurf eines Software-Systems an nichttrivialen praxisnahen Problemstellungen trainiert werden, und so das Wissen aus der Veranstaltung über Software-Design angewandt und gefestigt werden.

Die soziale Kompetenz, Teamfähigkeit und Eigenverantwortung wird durch selbständige Projektplanung und Projektorganisation einschließlich Aufgabenaufteilung, Zeitplanung und Aufwandsschätzung trainiert. Die Teamfähigkeit und die Kommunikationsfähigkeit werden gestärkt. Durch die Verwendung von fertigen Teilsystemen, Bibliotheken und Fremdsoftware, und auch dem Einsatz von Sprachen, die nicht intensiv in den Programmiersprachvorlesungen behandelt werden (Ruby, Python, ...) wird das selbständige Einarbeiten in neue Umgebungen und Systeme trainiert und gefördert.

- Sie verstehen die spezifischen Charakteristika und Herausforderungen von Projekten (z.B. im Unterschied zu Prozessen)
- Sie können die wichtigsten Projektmanagement-Tools je Projektphase anwenden (Initiierung, Planung, Durchführung, Abschluss)
- Sie können wesentliche Konzepte und Methoden anwenden, um Mensch-bezogene Herausforderungen im Projektumfeld zu analysieren und zu lösen (z.B. Motivation, Feedback, Veränderung)
- Sie können geeignete Projektmanagement-Ansätze (Wasserfall vs. Agil) je nach Projekttyp auswählen

Voraussetzungen und Empfehlungen:

- Vertrautheit mit den grundlegenden Konzepten und Phasen des Projektmanagements (Initiierung, Planung, Durchführung, Abschluss).
- Verständnis der verschiedenen Projektmanagementmethoden (z.B. Wasserfall, Agile).
- Fähigkeit zur Motivation und zum flexiblen Reagieren auf Änderungen.
- Grundlegende Kenntnisse in der Informatik und objektorientierten Programmier Techniken, insbesondere im Kontext von Software-Projekten.

Literatur:

Uwe Schmidt: Software-Projekt: Organisation und Themen, Unterlagen im Web: <http://www.fh-wedel.de/~si/praktika/SoftwarePraktikum/index.html>

- 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:

- Informatik Bachelor of Science
- Medieninformatik Bachelor of Science
- Wirtschaftsinformatik 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 Auslandssemester 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

◆ MB326 – Seminar Informatik

Verantwortliche:	Dennis Säring
Moduldauer:	6 Monate
Unterrichtssprache:	Deutsch

Bestandteile:

Teilleistung:	TB317 – Wissenschaftliches Arbeiten	Teilleistung:	TB318 – Seminar
Lernform:	Vorlesung	Lernform:	Seminar
Prüfungsform:	Abnahme	Prüfungsform:	Schriftl. Ausarbeitung (ggf. mit Präsentation)
Prüfungsdauer:	15 Min.	Prüfungsdauer:	20 Min.
ECTS:	1,0	ECTS:	4,0
Benotung:	Bestanden/nicht Bestanden	Benotung:	Drittnoten
Turnus:	jedes Semester	Turnus:	jedes Semester
Dauer (pro Termin):	2 Semesterwochenstunden	Dauer (pro Termin):	2 Semesterwochenstunden
Termine im Semester:	12 Termine	Termine im Semester:	12 Termine
Zeit in Veranstaltungen:	15 Stunden	Zeit in Veranstaltungen:	15 Stunden
Sonstiger Arbeitsaufwand		Sonstiger Arbeitsaufwand	
während Vorlesungszeit:	0 Stunden	während Vorlesungszeit:	0 Stunden
Aufwand während Semesterferien:	0 Stunden	Aufwand während Semesterferien:	0 Stunden
Flexibel einteilbarer Aufwand:	27 Stunden	Flexibel einteilbarer Aufwand:	90 Stunden
Gesamtaufwand:	Gesamtaufwand 37 Stunden weicht signifikant von erwarteten 1,0 ECTS ab	Gesamtaufwand:	Gesamtaufwand 103 Stunden weicht signifikant von erwarteten 4,0 ECTS ab
Lehrende:	Dennis Säring	Lehrende:	Dennis Säring

Studieninhalte:

Fachvorträge mit anschließender Gruppendiskussion.

Lernergebnisse:

Seminararbeiten dienen insbesondere dem Erlernen von Fertigkeiten zum Erstellen der Bachelor-Thesis. Wesentlich ist die eigenständige Erarbeitung und Darlegung der Inhalte zu einem vorgegebenen Thema unter Einhaltung der Formalia. Die Ausarbeitung soll das Interesse an einer eigenständigen Befassung mit Inhalten aus dem Themengebiet und den Einstieg in die zugehörige wissenschaftliche Fachliteratur und Methodik fördern und anregen. Schließlich ist die obligatorische Präsentation der Ergebnisse ebenfalls Aufgabe innerhalb des Seminars.

Nach erfolgreicher Teilnahme können sie ...

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

Voraussetzungen und Empfehlungen:

Grundlagen des Studienganges sollten gefestigt und ein umfassender Überblick über Spezialgebiete vorhanden sein, um die eigene Fähigkeit, ein bestimmtes, vorgegebenes Thema gemäß den Zielsetzungen des Seminars vorbereiten und zu präsentieren, beurteilen zu können.

Literatur:

keine Angabe

Studiengänge:

- Informatik Bachelor of Science

◆ MB150 – Bachelor-Thesis

Verantwortliche:	Sergei Sawitzki
Moduldauer:	6 Monate
Unterrichtssprache:	deutsch

Bestandteile:

Teilleistung:	BTB – Bachelor-Thesis
Lernform:	Thesis
Prüfungsform:	Abschlussarbeit
Prüfungsdauer:	30 Min.
ECTS:	12,0
Benotung:	Zeilchnoten
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:	Gesamtaufwand 23 Stunden weicht signifikant von erwarteten 3,0 ECTS ab
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:	1,50
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 Themenkomplex
- 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