



# **Modulhandbuch Bachelor Software-Design**

Fakultät Angewandte Informatik  
Prüfungsordnung ---  
Stand: 05.09.2024 11:39

## Inhaltsverzeichnis

- SD-01 Mathematik 1
- SD-02 Programmierung 1
- SD-03 Grundlagen der Informatik
- SD-04 Betriebssysteme und Netzwerke
- SD-05 Digitaltechnik und Rechnerorganisation
- SD-06 Schlüsselqualifikation 1
- SD-07 Mathematik 2
- SD-08 Programmierung 2
- SD-09 Algorithmen und Datenstrukturen
- SD-10 Internettechnologien
- SD-11 Betriebssysteme und Netzwerke 2
- SD-12 Schlüsselqualifikation 2
- SD-13 Datenbanken
- SD-14 Stochastik
- SD-15 Projektmanagement
- SD-16 Moderne Programmierparadigmen
- SD-17 HCI (Human Computer Interaction)
- SD-18 Schlüsselqualifikation 3
- SD-19 Software Engineering
- SD-20 Datenbanken 2
- SD-21 Cloud Computing
- SD-22 IT Sicherheit
- SD-23 FWP 1 (aus Softwaretechnologie)
- SD-24 Schlüsselqualifikation 4
- SD-25 Praxismodul
- SD-26 SE Projekt 1
- SD-27 Informatik-Seminar
- SD-28 Systems Engineering
- SD-29 Software Engineering 2
- SD-30 FWP 2
- SD-31 Schlüsselqualifikation 5
- SD-32 SE Projekt 2
- SD-33 FWP 3



SD-34 Wahlpflichtmodul 4 Unternehmensgründung / Compliance, Datenschutz  
SD-35 Bachelormodul



## SD-01 Mathematik 1

Modul Nr.	SD-01
Modulverantwortliche/r	Prof. Dr. Thorsten Matje
Kursnummer und Kursname	SD-01 Mathematik 1
Lehrende	Prof. Dr. Thorsten Matje
Semester	1
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Die Studierenden verfügen über grundlegendes allgemeines Wissen und grundlegendes Fachwissen im Bereich Informatik.

Im Einzelnen haben die Studierenden nach Abschluss des Moduls folgende Lernergebnisse erreicht:

#### Fachkompetenz

- Kenntnis und Verständnis von wesentlichen Grundlagen der Informatik, deren Konzepten und Methoden
- Fachliche Kompetenz diese Grundlagen selbständig nachzuvollziehen und an Beispielen anzuwenden



## Methodenkompetenz

- Formale Beweise durchführen, schriftlich und mit geeigneter Software
- Syntax von symbolischen Ausdrücken formal beschreiben
- Reguläre Ausdrücke mit endlichen Automaten implementieren
- Digitale Schaltkreise entwickeln

## Persönliche Kompetenz

- Studierende formulieren eigenständig logisch stichhaltige Argumente
- Studierende finden die Lücken in fehlerhaften Argumenten
- Studierende erkennen die Vor- und Nachteile der Digitalisierung

## Verwendbarkeit in diesem und in anderen Studiengängen

Dieses Modul ist Grundlage für die weiteren Informatik-Fächer. Es kann in anderen Informatik-Studiengängen verwendet werden.

## Zugangs- bzw. empfohlene Voraussetzungen

Keine Voraussetzungen.

## Inhalt

- Grundlagen der theoretischen Informatik
  - Logik
  - Berechenbarkeit
  - Endliche Automaten
  - Formale Sprachen
  - Komplexitätstheorie
- Grundlagen der technischen Informatik:
  - Schaltnetze und Schaltwerke
  - Rechnerarchitektur
  - Speicherorganisation
  - Internettechnologie

## Lehr- und Lernmethoden

- Seminaristischer Unterricht
- Bei jedem Thema werden entsprechende Software-Werkzeuge eingeführt und für die Übungen benutzt.
- Leistungsnachweis über Software-Werkzeuge



## Empfohlene Literaturliste

- Jon Barwise und John Etchemendy: Sprache, Beweis und Logik , Band I, Mentis 2005
- Susan H. Rodger und Thomas W. Finley: JFLAP: An Interactive Formal Languages and Automata Package , online bei <http://jflap.org/>
- Erich Hehner: Digital Circuit Design , Vorlesungsskript online bei <http://www.cs.toronto.edu/~hehner/DCD/DCD.pdf>
- J. Glenn Brookshear und Dennis Brylow: Computer Science--An Overview , 12th Ed, Pearson, 2015



## SD-02 Programmierung 1

Modul Nr.	SD-02
Modulverantwortliche/r	Prof. Dr. Andreas Berl
Kursnummer und Kursname	SD-02 Programmierung 1
Lehrende	Prof. Dr. Andreas Berl
Semester	1
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	ÜbL, schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Die Studierenden verfügen über grundlegendes allgemeines Wissen und grundlegendes Fachwissen im Bereich der Programmierung. Der Fokus liegt noch stark auf imperativer Programmierung, aber es werden auch erste objektorientierte Konzepte vermittelt. Die Studierenden sind in der Lage das Wissen praktisch anzuwenden und einfache bis mittelschwere Probleme zu lösen.

Im Einzelnen haben die Studierenden nach Abschluss des Moduls folgende Lernziele erreicht:

#### **Fachkompetenz**

Die Studierenden verstehen die Konzepte der modularen Gestaltung von Software.



### **Sozialkompetenz**

Im Rahmen der Vorlesungen finden Programmierübungen statt. Die Studierenden sind damit in der Lage, die Inhalte von Programmen ihrer Kollegen zu verstehen, zu kritisieren und durch eigene Programme zu komplementieren. Sie sind in der Lage, Programme in einer Form zu erstellen, die eine Kooperation im Team zulässt.

### **Methodenkompetenz**

Die Studierenden haben die Fähigkeit Programme unter Einsatz einer modernen objektorientierten Programmier-Plattform zu erstellen.

### **Persönliche Kompetenz**

Die Studierenden können eigene softwaretechnische Ideen umsetzen und gegenüber konkurrierenden Ansätzen verteidigen.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Grundlegende Einführung in die Programmierung

## **Zugangs- bzw. empfohlene Voraussetzungen**

Keine Voraussetzungen.

## **Inhalt**

### **Teil 1: Schnelleinstieg in die Imperative Programmierung**

- Überblick
  - Hallo Welt
  - Variablen, Abbildung im Arbeitsspeicher
  - Datentypen
  - Operatoren
- Kontrollstrukturen
  - Verzweigungen
  - Schleifen
- Programmierung
  - Programmiersprachen, Maschinensprache vs. Hochsprachen
  - Compiler
  - Programmerstellung
  - Compilerfehler vs. Laufzeitfehler
- Funktionen und Methoden
  - Rückgabewert, Name und Parameterliste
  - Rekursion
- Arrays
- Darstellung von Algorithmen



## Teil 2: Objektorientierte Programmierung

- Abstraktion
  - Klassen und Objekte
  - Instanzvariablen, Klassenvariablen, lokale Variablen
  - Methoden und Überladung
  - Konstruktoren
- Datentypen und Operatoren
  - Primitive Datentypen
  - Boolesche Operatoren
  - Bitweise Operatoren
  - Referenzdatentypen
  - Zuweisung
  - Object
  - Operatoren
  - Unterschiede zwischen Datentypen
  - Zuweisung, Kopie, Vergleiche
  - Parameterübergabe
  - Cast
  - Spezielle Referenzdatentypen
  - String, Array
  - Wrapper, Enum
- Kapselung
  - Abstrakte Datentypen
  - Geheimnisprinzip und Modularisierung
  - Modifikatoren
  - JavaDoc
  - Packages
- Vererbung
  - Überblick Vererbung in Java
  - Polymorphismus und Dynamische Bindung

## Teil 3: Weitere grundlegende Konzepte

- Zeichen, Bits und große Zahlen
  - Zeichen und Zeichenketten
  - Ein- und Ausgaben auf der Kommandozeile
  - Anwendung von Bitoperationen
  - Kleine und Große Zahlen
  - Die Klasse Math und Zufallszahlen
- Exceptions Fehlersuche und Testen
  - Exceptions
  - Fehler und Fehlersuche
  - Testen von Java Programmen
  - Junit



- Test Driven Development

## Lehr- und Lernmethoden

- Vorlesung mit PowerPoint
- Praktikum mit vielen Übungsaufgaben
- Gruppenarbeit
- Übungen, einschließlich Rechnerübungen (mit Leistungsnachweis)

## Besonderes

keine

## Empfohlene Literaturliste

Java-Programmierung: Das Handbuch zu Java 8

Guido Krüger, Heiko Hansen

O'Reilly Verlag Köln

8. Auflage 2014

ISBN 978-3-95561-514-7

Handbuch der Java-Programmierung

Guido Krüger, Heiko Hansen

7. Auflage 2011

HTML-Ausgabe 7.0.0 · © 1998, 2011

[http:// www.javabuch.de/download.html](http://www.javabuch.de/download.html)

Java ist auch eine Insel: Einführung, Ausbildung, Praxis

Christian Ullenboom

Rheinwerk Computing

16. Auflage 2021

ISBN 978-3-8362-8745-6

Java ist auch eine Insel: Einführung, Ausbildung, Praxis

Christian Ullenboom

Rheinwerk Computing

15. Auflage 2019

[http:// openbook.rheinwerk-verlag.de/javainsel](http://openbook.rheinwerk-verlag.de/javainsel)



## SD-03 Grundlagen der Informatik

Modul Nr.	SD-03
Modulverantwortliche/r	Prof. Dr. Thorsten Matje
Kursnummer und Kursname	SD-03 Grundlagen der Informatik
Lehrende	Prof. Dr. Thorsten Matje
Semester	1
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	ÜbL, schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Die Studierenden verfügen über grundlegendes allgemeines Wissen und grundlegendes Fachwissen im Bereich Informatik.

Im Einzelnen haben die Studierenden nach Abschluss des Moduls folgende Lernergebnisse erreicht:

#### Fachkompetenz

- Kenntnis und Verständnis von wesentlichen Grundlagen der Informatik, deren Konzepten und Methoden
- Fachliche Kompetenz diese Grundlagen selbständig nachzuvollziehen und an Beispielen anzuwenden



### **Methodenkompetenz**

- Formale Beweise durchführen, schriftlich und mit geeigneter Software
- Syntax von symbolischen Ausdrücken formal beschreiben
- Reguläre Ausdrücke mit endlichen Automaten implementieren
- Digitale Schaltkreise entwickeln

### **Persönliche Kompetenz**

- Studierende formulieren eigenständig logisch stichhaltige Argumente
- Studierende finden die Lücken in fehlerhaften Argumenten
- Studierende erkennen die Vor- und Nachteile der Digitalisierung

### **Verwendbarkeit in diesem und in anderen Studiengängen**

Dieses Modul ist Grundlage für die weiteren Informatik-Fächer. Es kann in anderen Informatik-Studiengängen verwendet werden.

### **Zugangs- bzw. empfohlene Voraussetzungen**

Keine Voraussetzungen.

### **Inhalt**

- Grundlagen der theoretischen Informatik
  - Logik
  - Berechenbarkeit
  - Endliche Automaten
  - Formale Sprachen
  - Komplexitätstheorie
- Grundlagen der technischen Informatik:
  - Schaltnetze und Schaltwerke
  - Rechnerarchitektur
  - Speicherorganisation
  - Internettechnologie

### **Lehr- und Lernmethoden**

- Seminaristischer Unterricht
- Bei jedem Thema werden entsprechende Software-Werkzeuge eingeführt und für die Übungen benutzt.
- Leistungsnachweis über Software-Werkzeuge



## Empfohlene Literaturliste

- Jon Barwise und John Etchemendy: Sprache, Beweis und Logik , Band I, Mentis 2005
- Susan H. Rodger und Thomas W. Finley: JFLAP: An Interactive Formal Languages and Automata Package , online bei <http://jflap.org/>
- Erich Hehner: Digital Circuit Design , Vorlesungsskript online bei <http://www.cs.toronto.edu/~hehner/DCD/DCD.pdf>
- J. Glenn Brookshear und Dennis Brylow: Computer Science--An Overview , 12th Ed, Pearson, 2015



## SD-04 Betriebssysteme und Netzwerke

Modul Nr.	SD-04
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-04 Betriebssysteme und Netzwerke
Lehrende	Prof. Dr. Andreas Wöfl
Semester	1
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

**Die Studierenden erwerben folgende fachliche Kompetenzen:**

#### **Teil Betriebssysteme**

Die Studierenden erhalten Einblick in die Bedeutung von Betriebssystemen als zentrale Grundlage für die Informationsverarbeitung in Unternehmen. Für die heutigen Ausprägungen von Betriebssystemen bauen sie Verständnis auf. Nach Absolvieren des Teilmoduls Betriebssysteme haben die Studierenden folgende Lernziele erreicht:

- Die Studierenden erlangen Kenntnis von Konzepten und Technologien, die für den Aufbau von Betriebssystemen notwendig sind und Wissen über den modularen Aufbau und die Funktionsweise von Betriebssystemen.



- Die Studierenden erwerben Wissen und Fertigkeiten über die Konfiguration, die Administration und die sichere Anwendung von Betriebssystemen anhand von kommerziellen Betriebssystemen.
- Die Studierenden ordnen und bewerten moderne Betriebsformen von Rechenzentren, wie z. B. Virtualisierung oder Cloud Computing im Kontext der Betriebssysteme.
- Die Studierende erhalten einen Einblick in die theoretischen Grundlagen eines Linuxsystems sowie einen Überblick über die wichtigsten Shellbefehle.
- Die Studierenden installieren und administrieren einen Linuxserver.

#### **Teil Netzwerke:**

- Die Studierenden lernen die Grundlagen sowie die physikalische und logische Anordnung von Geräten in einem Computernetzwerk.
- Die Studierenden bewerten Netzwerktopologien anhand graphentheoretischer Eigenschaften.
- Die Studierenden erwerben Wissen über den Aufbau und die Funktionsweise des Internet.
- Die Studierenden sind in der Lage anhand gegebener Netzwerkparameter die wichtigsten Performance-Kennzahlen wie Durchsatz oder Verzögerung zu berechnen.
- Die Studierenden erkennen die Bedeutung von Schichtenmodellen und können Aufgaben und Funktionen den Schichten des ISO/OSI Modells zuordnen.
- Die Studierenden erlangen Kenntnis über die wichtigsten Netzwerkprotokolle wie z.B. Ethernet, TCP, IP, DNS und können die Konzepte der jeweiligen Protokolle nachvollziehen und erklären.
- Die Studierenden können einfache Netzwerkanwendungen mit Sockets programmieren.

### **Verwendbarkeit in diesem und in anderen Studiengängen**

Dieses Modul ist Grundlage für die weiteren Informatik-Fächer.

### **Zugangs- bzw. empfohlene Voraussetzungen**

Keine Voraussetzungen.

### **Inhalt**

#### **Teil Betriebssysteme**

#### **Theoretische Inhalte**



- Rechtenmanagement (Authentifizierung, Authorisierung)
- Prozesse & Threads, Inter-Prozess Kommunikation
- Deadlocks, Mutex-Verfahren
- Peripherie / Ein-/Ausgabe
- Betriebssystem API, Userspace / Kernelspace

### **Praktische Inhalte**

- Umgang mit Linux / Unix / POSIX
- Umgang mit Shells - graphisch und textbasiert (insbesondere praktischer Umgang mit der Kommandozeile)
- Nutzung von Systemvirtualisierung (z.B.: Hypervisors, VirtualBox, XEN, Docker, ...)
- Verwendung von Systemcalls

### **Teil Netzwerke**

#### **Theoretische Inhalte**

- Schichtenmodell: OSI
- Netzwerktopologien (Bus, Baum, Stern, teil-/vollvermascht)
- Anwendungsschicht: HTTP, SMTP & IMAP, DNS
- Transportschicht: Sockets, UDP, TCP
- Ausblick auf die Netzwerkschicht: IPv4/v6

#### **Praktische Inhalte**

- Verwendung von Werkzeugen und Techniken zur Netzwerkanalyse und -konfiguration (z.B. Ping, Traceroute, PuTTY/telnet, nslookup, ...)
- Verwendung von Browser Debugging Tools (Netzwerkconsole, ...)
- Textbasierte Anwendungsprotokolle verstehen und umsetzen (z.B. HTTP Interaktionen)

### **Lehr- und Lernmethoden**

Seminaristischer Unterricht mit praktischen Übungen

### **Empfohlene Literaturliste**

#### **Teil Betriebssysteme**

- Andrew S. Tanenbaum, Herbert Bos; Modern Operating Systems; Prentice Hall, 4th ed., 2014
- Evi Nemeth, Garth Snyder, Trent R. Hein et al.; Unix and Linux System Administration Handbook, Addison-Wesley, 5th ed., 2018
- Micha Gorelick & Ian Ozsvald; High Performance Python; O'Reilly, 2014

#### **Teil Netzwerke**



- James F. Kurose, Keith F. Ross; Computer Networking: A Top-Down Approach; Pearson, 7th ed., 2017
- Andrew S. Tanenbaum, David J. Wetherall; Computer Networks; Pearson, 5th ed., 2014



## SD-05 Digitaltechnik und Rechnerorganisation

Modul Nr.	SD-05
Modulverantwortliche/r	Prof. Dr. Robert Bösnecker
Kursnummer und Kursname	SD-05 Digitaltechnik und Rechnerorganisation
Lehrende	Prof. Dr. Robert Bösnecker
Semester	1
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

### Zugangs- bzw. empfohlene Voraussetzungen

keine

### Inhalt

#### Einführung in die Digitaltechnik

- Grundlagen der Digitaltechnik
  - Definition und Bedeutung



- Unterschied zwischen analoger und digitaler Technik
- Digitale Darstellung von Text, Sprache und Bildern
  - ASCII Text und Steuerzeichen
  - Sampling von Sprache und Klängen
  - Image Digitalisierung und Speicher-Abbild

### **Logikschaltungen**

- Grundlegende Logikgatter
  - AND, OR, NOT, NAND, NOR, XOR, XNOR
  - Technischer Aufbau von Gattern
- Kombinatorische Logik
  - Zusammenfassung Boolesche Algebra
  - Minimierung, DNF
  - Komplexe Bausteine (TTL ICs)
- Schaltungsdesign
  - Addierer, Subtrahierer, Multiplexer, Demultiplexer, De/Encoder, Parity

### **Sequentielle Schaltungen**

- Grundlagen
  - Unterschied zwischen kombinatorischen und sequentiellen Schaltungen
- Flip-Flops und Register
  - SR-, D-, JK-, T-Flip-Flops
  - Schieberegister, Zähler
- Finite State Machines (FSM)
  - Zustandsdiagramme, FSMs
  - Medwedew-, Mealy- und Moore-Automaten
  - VHDL Sprache und Automatentheorie

### **Speichertechnologien**

- Arten von Speicher
  - RAM (SRAM, DRAM), ROM (PROM, EPROM, EEPROM)
  - Eigenschaften und Grenzen
- Cache-Speicher
  - Funktionsweise und Bedeutung bei uProzessoren
  - Cache-Architektur und Cache-Kohärenz

### **Rechnerorganisation**

- Grundlegende Architekturprinzipien
  - Von-Neumann-Architektur vs. Harvard-Architektur
- CPU-Organisation
  - ALU, Register, Kontrollwerk
  - Befehlssatzarchitekturen (RISC vs. CISC)



- Pipeline-Architektur
  - Grundkonzepte der Pipelining
  - Hazards und deren Behebung, Branch Prediction

### **Eingabe-/Ausgabe-Techniken**

- Grundlagen der I/O-Techniken
  - I/O-Module, serielle und parallele Schnittstellen
- Direkter Speicherzugriff (DMA)
  - Funktionsweise und Vorteile

### **Systembusse und Kommunikation**

- Bus-Systeme
  - Adress-, Daten- und Steuerbus
- Bus-Architekturen
  - PCI, USB, SATA, SPI, I2C, I2S, etc.

### **Mikrocontroller und eingebettete Systeme**

- Grundlagen der Mikrocontroller
  - Aufbau und Funktion
- Programmierbare Logik
  - FPGA, CPLD
- System-on-Chip (SoC)
  - Einführung in SoC-Architekturen.
  - Integration von Prozessoren, Speicher, und I/O in einem Chip.
  - Überblick über die Entwicklung von FPGA-basierten Systemen.

### **Fehlererkennung und -korrektur**

- Grundlagen
  - Paritätsbit, Hamming-Code
- Fehlerkorrigierende Codes
  - ECC (Error-Correcting Code)

### **Moderne Entwicklungen und Trends**

- Multicore- und Parallelverarbeitung
- Rekonfigurierbare Architekturen
- Neuromorphe Systeme
- Rechenarchitekturen für Künstliche Intelligenz (KI) und Maschinelles Lernen (ML)
  - Tensor Processing Units (TPUs) und Grafikprozessoren (GPUs)

### **Digital Signal Processing (DSP) Grundlagen:**

- Einführung in die digitale Signalverarbeitung
- Diskrete Fourier-Transformation (DFT), Fast Fourier-Transformation (FFT)
- Anwendung von DSP in der Kommunikationstechnik, Audioverarbeitung, Bildverarbeitung



### **Leistung und Energieeffizienz in digitalen Systemen:**

- Prinzipien des Energiemanagements in digitalen Schaltungen.
- Low-Power Design Techniken (z.B. Clock-Gating, Power-Gating).
- Relevanz von Energieeffizienz in mobilen und eingebetteten Systemen.

### **Simulation und Verifikation in der Digitaltechnik**

- VHDL Anwendung
  - Behavioural Modelling
  - Fitting, CPLD und FPGA
  - Simulation und Fehlerkorrektur
  - JEDEC Datei und Download-Tools

### **Lehr- und Lernmethoden**

- Seminaristischer Unterricht mit Übungen
- Unterstützung durch die E-Learning-Plattform und Lehrvideos



## SD-06 Schlüsselqualifikation 1

Modul Nr.	SD-06
Modulverantwortliche/r	Prof. Dr. Roland Zink
Kursnummer und Kursname	SD-06 Schlüsselqualifikation 1
Lehrende	Prof. Dr. Thomas Geiß N.N. Prof. Dr. Roland Zink
Semester	1
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Der Umstieg von der Schule zu Hochschule stellt viele Studierende gleich zu Beginn ihres Studiums vor Herausforderungen. Weg von vorgegebenen Stundenplänen und Lehrplanbezug, hin zu Eigen- und Selbstständigkeit sowie Verantwortung. Das Modul Schlüsselqualifikation 1 soll auf diese Herausforderungen insbesondere auch mit Blick auf die Digitalisierung und den wirtschaftlichen Bezug (Betriebspraktikum im 5. Semester) vorbereiten. Die Lernergebnisse des Moduls setzen sich folglich aus den beiden Fächern "Betriebswirtschaft" (**Fach A**) und "Medienkompetenz und Selbstorganisation" (**Fach B**) zusammen.



## **Fach A**

Im Fach Betriebswirtschaft setzen sich die Studierenden insbesondere mit der Allgemeinen BWL, der Kosten- und Leistungsrechnung sowie dem Personalmanagement auseinander. Obwohl die Studierenden einen technischen bzw. informatikorientierten Studiengang belegen, soll durch das angeeignete betriebswirtschaftliche Wissen der Berufseinstieg erleichtert werden. Durch die Verbreiterung der Wissensbasis bei den Studierenden sollen suboptimale Entscheidungen in Unternehmen vermieden werden.

### **Fachkompetenz**

- Die Studierenden lernen die betrieblichen Funktionalbereiche im Überblick und ausgewählte Konzepte der Unternehmensführung/Strategieentwicklung kennen.
- Die Studierenden kennen und verstehen die Grundsätze und Methoden einer systematischen Entscheidungsfindung.
- Die Studierenden kennen die Zwecke der Kosten- und Leistungsrechnung (KLR) und den Aufbau eines KLR-Systems
- Sie sind mit wichtigen Instrumenten der KLR, der Kostenstellen- und Kostenträgerrechnung sowie der kurzfristigen Erfolgsrechnung vertraut
- Sie werden befähigt, kostenstellen- und auftragsbezogene Soll-IstVergleiche (SIV) durchzuführen und bewerten
- Sie können die Teilkostenrechnung in Form der Deckungsbeitragsrechnung anwenden
- Sie werden befähigt, Entscheidungsrechnungen auf Basis der KLR durchzuführen

## **Fach B**

Das Fach Selbstorganisation und Medienkompetenz gliedert sich inhaltlich in drei große Blöcke. Der erste Block beinhaltet eine gute und dem Studienzweck angepasste Selbstorganisation mit der Einführung in die neue Herausforderung des Studiums, dem Zeitmanagement und der Lernumgebung der THD. Den zweiten Block bildet Medienkompetenz, indem insbesondere Aspekte der digitalen Transformation unserer Gesellschaft aufgegriffen werden. Neben den Inhalten des Medienkompetenzrasters der Kultusministerkonferenz (2016) mit seinen sechs Säulen: 1) Suchen, Verarbeiten und Aufbewahren, 2) Kommunizieren und Kooperieren, 3) Produzieren und Präsentieren, 4) Schützen und sicher Agieren, 5) Problemlösen und Handeln und 6) Analysieren und Reflektieren werden studienorientiert u.a. der Umgang mit wissenschaftlichen Statistiken und Literatur, Fake News, Plagiate, Datenschutz, Urheberrechte und Formen der Wissenschaftskommunikation thematisiert. Der dritte Block vermittelt Einblicke in den wissenschaftlichen Umgang mit Daten. Inhalte hierzu sind Datenerhebung, -auswertung und -visualisierung sowie Forschungsdaten- und Wissensmanagement.

### **Fachkompetenz**

- Die Studierenden kennen verschiedene digitale Medien zur Lernorganisation (insb. das Angebot der THD) und können diese anwenden.



- Die Studierenden werden befähigt, sowohl analoge als auch digitale Lehr- und Lerninhalte gezielt für ihr Studium auszuwählen.
- Die Studierenden sind befähigt, mit digitalen Medien kompetent und zielgerichtet umzugehen.
- Die Studierenden können ihr Studium zeitlich wie inhaltlich organisieren und die Informationsfülle zielgerichtet bearbeiten.
- Die Studierenden kennen die Grundlagen zur Arbeit mit wissenschaftlichen Quellen (v.a. Statistiken und Literatur) und können studiengangorientiert damit arbeiten.
- Die Studierenden erhalten einen Einblick in die verschiedenen Formen der Wissenschaftskommunikation und kennen Regeln des wissenschaftlichen Arbeitens bzw. Folgen wissenschaftlichen Fehlverhaltens.
- Die Studierenden wissen, was Daten, Information und Wissen sind und lernen den Umgang mit Forschungsdaten bzw. Daten im Studium.

## **Fach A und B**

### **Methodenkompetenz**

- Die Studierenden werden in der KLR zu einem transparenz-, struktur- und entscheidungsorientierten Arbeiten befähigt
- Den Studierenden wird bewusst, dass die KLR zweckorientiert zu konzipieren ist.
- Die Studierenden werden zu selbstständigen Arbeiten befähigt.
- Die Studierenden erwerben Kompetenzen beim Umgang mit digitalen Medien und wissenschaftlichen Daten.
- Die Studierenden erlernen Strategien der Wissensaneignung mit Blended Learning Verfahren.

### **Persönliche Kompetenz**

- Die Studierenden erlernen durch Übungen selbstständige und problem-, lösungs- bzw. handlungsorientiertes Arbeiten.

### **Sozialkompetenz**

- Die Studierenden trainieren in den Übungen Partner- und Teamarbeit.
- Die Studierenden erlernen eigenverantwortliches Arbeiten

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Das Modul legt Grundlagen für das Studium im Allgemeinen und ist insbesondere mit folgendem weiterführenden Modul verknüpft:

AI-B: Schlüsselqualifikation 2

KI-B und CY-B: Schlüsselqualifikation 3

KI-B und CY-B: Schlüsselqualifikation 4

AI-B, KI-B und CY-B: Praxismodul



AI-B, KI-B und CY-B: Bachelormodul Studiengang:  
(BA Angewandte Informatik, BA Cyber Security und BA Künstliche Intelligenz)

## **Zugangs- bzw. empfohlene Voraussetzungen**

Keine Voraussetzungen.

## **Inhalt**

### **Fach A**

- Das Unternehmen im Überblick
- Unternehmensführung und Unternehmenspolitik
- Vision, Ziele, Strategien
- Konstitutive Unternehmensentscheidungen
- Produktionsfaktoren
- Betriebliche Funktionen
- Überblick über die Ansätze der Entscheidungstheorie
- Zwecke der KLR u. Kostenzuordnungsprinzipien
- Systeme der KLR
- Spezifische kostenrechnerische Inhalte in den Bereichen KI und CS
- Die KLR auf der Vollkostenbasis
- Kostenartenrechnung
- Kostenstellenrechnung
- Kostenträgerrechnung
- Die KLR auf Teilkostenbasis (Deckungsbeitragsrechnung)
- Die kurzfristige Erfolgsrechnung
- Entscheidungsorientierte KLR inkl. des Grundsatzes der relevanten Kosten

### **Fach B**

- Neue Herausforderung Studium: kritisch und reflektiert sein
- Selbstorganisation und Zeitmanagement
- Die Lernumgebung THD und Studium gestalten
- Medienkompetenz: Digitale Medien im studentischen Lernkontext
- Statistiken und Literatur für wissenschaftliche Zwecke
- Fake News, Plagiate sowie Urheber- und Nutzungsrechte im wissenschaftlichen Kontext
- Wissenschaftskommunikation: Digitale Medien in der Wissenschaft und Kommunikation
- Daten, Information und Wissen



- Wissenschaftliche Daten auswerten und visualisieren
- Forschungsdatenmanagement
- Wissensmanagement

## Lehr- und Lernmethoden

- Seminaristischer Unterricht mit Gruppen- und Partnerarbeit
- Projektarbeit
- Blended Learning

## Besonderes

keine

## Empfohlene Literaturliste

### Fach A

- Däumler K., Grabe J. (2013): Kostenrechnung 1 ? Grundlagen, 11. Aufl., NWB-Verlag, Herne.
- Dörsam, P. (2013): Grundlagen der Entscheidungstheorie anschaulich dargestellt, 6. Auflage, PD-Verlag, Heidenau.
- Friedl G., Hofmann Ch., Pedell B. (2017): Kostenrechnung: Eine entscheidungsorientierte Einführung, 3. Aufl., Vahlen Verlag, München.
- Jorasz W., Baltzer B. (2019): Grundlagen der Kosten- und Leistungsrechnung: Lehrbuch mit Aufgaben und Lösungen, SchäfferPoeschel Verlag, Stuttgart.
- Wöhe, G. (2016), Einführung in die allgemeine Betriebswirtschaftslehre, 26. Auflage, Vahlen, München.

### Fach B

- Gapski, H., Oberele, M. & Staufer, W. (Hrsg.) (2017): Medienkompetenz. Herausforderung für Politik, politische Bildung und Medienbildung. Bonn. Dieses Buch steht zum kostenlosen Download zur Verfügung: <https://www.bpb.de/lernen/digitale-bildung/medienpaedagogik/medienkompetenz-schriftenreihe/>
- Lehner, F. (2021): Wissensmanagement. Grundlagen, Methoden und technische Unterstützung. 7. Auflage. München.
- Voss, R. (2014): Wissenschaftliches Arbeiten. 3. Auflage. Wien. (Über die THD-Bibliothek als eBook erhältlich)
- (Zusätzlich werden Internetdokumente und Leitfäden verwendet!)



## SD-07 Mathematik 2

Modul Nr.	SD-07
Modulverantwortliche/r	Prof. Dr. Thorsten Matje
Kursnummer und Kursname	SD-07 Mathematik 2
Lehrende	Prof. Dr. Thorsten Matje
Semester	2
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Die Studierenden erwerben vertiefte Kenntnisse mathematischer Themen, die in Anwendung in der Informatik und in mathematischen Gebieten von Bedeutung sind oder die zur vertieften Abrundung mathematischer Grundkonzepte notwendig sind. Der Fokus liegt dabei auch auf mathematischen Denk-, Arbeits- und Modellierungsmethoden. Die Studierenden sind in der Lage mathematische Fragestellungen aus der Informatik zu erkennen, zu modellieren und zu lösen. Die zugehörigen algorithmischen Methoden der Mathematik werden exemplarisch erarbeitet. Die Studierenden sind in der Lage weiterführende Veranstaltungen mit mathematischer Modellbildung erfolgreich zu absolvieren.



Im Vordergrund steht die Fach- und die Methodenkompetenz in den behandelten Themenfeldern.

Der Erwerb von sozialen Kompetenzen steht bei diesem Modul naturgemäß nicht im Vordergrund, wird aber durch Kooperation der Studierenden und gemeinsames Erarbeiten von Lösungen gefördert.

Die persönliche Kompetenz wird durch vertieftes selbständiges Erarbeiten und Lösen komplexer Probleme gefördert. Durch die Anwendung mathematischer Lösungstechniken und deren kritische Durchdringung erarbeiten sich die Studierende die Fähigkeit zum abstrakten und analytischen Denken.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Die Studierenden sind in der Lage weiterführenden Veranstaltungen mit mathematischer Modellbildung erfolgreich zu absolvieren.

Weiter kann das Modul für weiterbildende, konsekutive und aufbauende Masterstudiengänge verwendet werden.

## **Zugangs- bzw. empfohlene Voraussetzungen**

Empfohlen:

- Inhalt des Moduls Mathematik 1

## **Inhalt**

- 1 Komplexe Zahlen und trigonometrische Funktionen
  - Geometrische Darstellung von komplexen Zahlen
  - Komplexe Potenzreihen und Anwendungen in der Trigonometrie
  - Kreisteilung
  - Fundamentalsatz der Algebra
  - Satz von DeMoivre
- 2 Zahlentheorie, Computeralgebra und Kryptographie
  - Teilbarkeit und Primzahlen
  - Division mit Rest
  - Euklidischer Algorithmus
  - Äquivalenzrelation und Äquivalenzklassen
  - Vertretersysteme
  - Gruppen und Ringe
  - Invertieren von Restklassen
  - Erweiterter Euklidischer Algorithmus
  - Chinesischer Restsatz
  - Die Euler'sche Phifunktion



- Kleiner Satz von Fermat
- Exponentiation im Restklassenring
- Faktorisierung von Zahlen
- Kryptographie
- RSA-Verfahren
- Digitale Signatur
- Hashfunktionen
- 3 Lineare Differentialgleichungen
  - Gewöhnliche Differentialgleichungen
  - Anfangswertprobleme
  - Trennbare Variablen
  - Substitution
  - Homogene lineare Differentialgleichungen 1. Ordnung
  - Inhomogene lineare Differentialgleichungen 1. Ordnung (Variation der Konstanten)
  - Anwendungsbeispiel: Radioaktiver Zerfall
- 4 Numerische Nullstellenberechnung
  - Bisektion
  - Sekantenverfahren
  - Newtonverfahren
  - Horner Schema

## Lehr- und Lernmethoden

Lehre im JITT-Format (Just-in-Time-Teaching), also Abbildung der Vorlesung durch interaktive Lehrvideos inkl. Lernkontrollen sowie verlinkte Literatur und Auswahl der vorzurechnenden Übung.

In der Präsenz werden die gelernten Inhalte mit Übungsaufgaben vertieft. Dabei wird jeweils eine Übung pro Thema vorgerechnet, und weitere Aufgaben werden von den Studierenden unter Anleitung selbst bearbeitet.

## Besonderes

Eine der 4 SWS wird als Übung im Computerraum in 2 Gruppen vom Dozenten angeboten.

## Empfohlene Literaturliste

Kapitel 1: Komplexe Zahlen

- Angewandte Mathematik mit Mathcad. Lehr- und Arbeitsbuch, Josef Trölb
- Komplexe Zahlen und ebene Geometrie, Joachim Engel



- Komplexe Zahlen, Jörg Kortemeyer
- Mathematische Grundlagen für die Natur- und Ingenieurwissenschaften, Michael Jung
- Elementare Technomathematik, Harald Schmid

#### Kapitel 2: Algebra

- Mathematische Geschichten IV Euklidischer Algorithmus, Modulo-Rechnung und Beweise, Susanne Schindler-Tschirner
- Moderne Verfahren der Kryptographie, Albrecht Beutelspacher
- Das RSA-Verfahren: Verschlüsseln und Entschlüsseln auf Basis der Algebra, Guido Walz
- Komplexitätstheorie und Kryptologie, Jörg Rothe

#### Kapitel 3: Differentialgleichungen

- Gewöhnliche Differentialgleichungen, Heidrun Günzel
- Mathematik für Ingenieurwissenschaften: Vertiefung, Harald Schmid
- Differentialgleichungen für Einsteiger, Thorsten Imkamp
- Mathematik für Ingenieure und Naturwissenschaftler, Wilhelm Merz

#### Kapitel 4: Numerik

- Fixpunkte und Nullstellen, Guido Walz



## SD-08 Programmierung 2

Modul Nr.	SD-08
Modulverantwortliche/r	Prof. Dr. Andreas Berl
Kursnummer und Kursname	SD-08 Programmierung 2
Lehrende	Prof. Dr. Andreas Berl
Semester	2
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	ÜbL, schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Das Ziel dieses Moduls ist es, den Studierenden fortgeschrittene Programmierkonzepte, Modellierungsmethoden, verschiedene Programmierparadigmen und verschiedene Werkzeuge zu vermitteln. Die Studierenden erwerben eine solidere Grundlage für den Entwurf und die Implementierung von Software. Sie lernen auch, wie man professionelle Software-Werkzeuge benutzt. Dadurch werden sie in der Lage sein, in Teams hochwertige Software zu schreiben.

Im Einzelnen haben die Studierenden nach Abschluss des Moduls folgende Lernergebnisse erreicht:

#### Fachkompetenz



- Die Studierenden verstehen die Konzepte der professionellen Erstellung von Software.
- Die Studierenden können eigene softwaretechnische Ideen umsetzen.

### **Methodenkompetenz**

- Die Studierenden haben die Fähigkeit, hochqualitative Programme unter Einsatz moderner Werkzeuge zu erstellen.

### **Persönliche Kompetenz**

- Die Studierenden können eigene softwaretechnische Ideen gegenüber konkurrierenden Ansätzen verteidigen.

### **Sozialkompetenz**

- Im Rahmen der Lehrveranstaltung finden Programmierübungen statt. Die Studierenden sind damit auch in der Lage, Programme anderer Studierenden zu verstehen, zu kritisieren und zu komplementieren.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Unter anderem:

- Software Engineering
- Sichere Programmierung
- Penetration Testing

## **Zugangs- bzw. empfohlene Voraussetzungen**

### **Zugangsvoraussetzungen:**

- keine spezifischen

### **empfohlene Voraussetzungen:**

- Inhalt des ersten Semesters, insbesondere Programmierung 1
- Grundlagen der Mathematik

## **Inhalt**

- Einführung: Wiederholung der grundlegenden Programmierkonzepte, Einführung in Python
- Werkzeuge: IDEs, interaktive Umgebungen, Jupyter-Notebooks, Revisionskontrolle, Debugger, Timing von Code, Profiler, Cython, Logger, Arbeitspaket-Tracker, Bugtracker, Build Chains
- Code-Konventionen: Styleguides, Clean Code
- Modellierung: Anwendungsfalldiagramme, Aktivitätsdiagramme, Klassendiagramme, Objektdiagramme
- OOP: Decorators, Refactoring, Entwurfsmuster
- Testen: Unit-Tests, testgetriebene Entwicklung, Testabdeckung



- Speicherverwaltung: Stack und Heap, manuelles Freigeben von Speicher, Garbage Collection, Interning
- Ausnahmebehandlung: Raising und Catching, Asserts
- Dateien: Lesen und Schreiben, Löschen, Serialisierung, JSON, pickle, tabellarische Daten
- Multithreading: Parallelism und Concurrency, Erstellen von Threads, Global Interpreter Lock (GIL)
- Logik-Programmierung: Logik, deklarative Programmierung, Prolog

## Lehr- und Lernmethoden

- Vorlesungen
- Diskussion von wissenschaftlichen Artikeln und aktuellen Nachrichten
- Übungen, einschließlich Rechnerübungen

## Empfohlene Literaturliste

- S. Chacon and B. Straub, " Pro Git ", Apress, 2nd edition, 2014.
- M. Goodrich et al., " Data Structures and Algorithms in Python ", John Wiley & Sons, 2013.
- C. Larman, " Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development ", 3rd edition, Prentice Hall, 2004.
- E. Matthes, " Python Crash Course: A Hands-On, Project-Based Introduction to Programming ", 2nd edition, 2019.



## SD-09 Algorithmen und Datenstrukturen

Modul Nr.	SD-09
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-09 Algorithmen und Datenstrukturen
Lehrende	Prof. Dr. Andreas Wöfl
Semester	2
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	ÜbL, schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Ziel dieses Moduls ist es, eine Einführung in eine der wichtigsten Grundlagen eines Informatikstudiums zu geben: Algorithmen und Datenstrukturen. Eine Datenstruktur ermöglicht es einem Programmierer, Daten in konzeptionell handhabbare Zusammenhänge zu strukturieren. Ein Algorithmus ist eine endliche Folge von wohldefinierten, computer-implementierbaren Anweisungen, um eine Klasse von Problemen zu lösen oder eine Berechnung durchzuführen. Algorithmen arbeiten oft mit Datenstrukturen. Dieser Kurs bietet eine Reise durch die Informatik. Die Studierenden erwerben eine solide Grundlage davon, wie die wichtigsten Algorithmen und Datenstrukturen funktionieren. Sie lernen auch, wie man effiziente Algorithmen und Datenstrukturen entwirft.



Im Einzelnen haben die Studierenden nach Abschluss des Moduls folgende Lernergebnisse erreicht:

### **Fachkompetenz**

- Die Studierenden verstehen die Konzepte der gängigsten Algorithmen und Datenstrukturen. (2 - Verstehen)

### **Methodenkompetenz**

- Die Studierenden haben die Fähigkeit, hochqualitative Programme unter Einsatz von Algorithmen und Datenstrukturen zu erstellen. (3 - Anwenden)

### **Persönliche Kompetenz**

- Die Studierenden können eigene Algorithmen und Datenstrukturen umsetzen und gegenüber konkurrierenden Ansätzen verteidigen. (6 - Erschaffen)

### **Sozialkompetenz**

- Im Rahmen der Lehrveranstaltung finden Programmierübungen statt. Die Studierenden sind damit auch in der Lage, Algorithmen und Datenstrukturen anderer Studierender zu verstehen, zu kritisieren und zu komplementieren. (5 - Beurteilen)

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Unter anderem:

- Software Engineering
- Assistenzsysteme
- Sprachverarbeitung
- Maschinelles Lernen
- Bildverstehen
- Deep Learning/Big Data

## **Zugangs- bzw. empfohlene Voraussetzungen**

Empfohlen:

- Inhalt des ersten Semesters, insbesondere Programmierung 1
- Grundlagen Mathematik

## **Inhalt**

- Einführung: Algorithmen-Definition, Klassifizierung von Algorithmen
- Graphen: Graphen-Definitionen, Anwendungen in der Informatik, Shortest Path, Lowest Cost, A\*



- Komplexitätsanalyse: Zeitkomplexität, O-, Omega-, Theta-, o- und O-Tilde-Kalküle, pseudo-polynomielle Komplexität, Speicherkomplexität
- Listen: Arrays, dynamische Arrays/Listen, Amortisierung, Basisoperationen, Stacks, Warteschlangen, verkettete Listen
- Rekursion: Suche, Divide and Conquer, Rekurrenzgleichungen, Master Theorem, Backtracking, dynamische Programmierung
- Sortierung: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quicksort, untere Schranken
- Bäume: Binärbäume, Traversieren, fortgeschrittene Arten von Bäumen, Entscheidungsbäume
- Maps und Hash-Tabellen: Key-Value-Speicher, Hashing, Kollisionsbehandlung
- Ausgewählte Themen: schnelle Matrizenmultiplikation, Zufallszahlengenerierung, schnelle inverse Quadratwurzel, Primzahlen, Bloom-Filter, Union-Find, Median der Mediane, String-Matching
- Quantencomputing: Qubits, Quantengatter, Quantencomputer, Quantenalgorithmen

## Lehr- und Lernmethoden

- Vorlesungen
- Diskussion von wissenschaftlichen Artikeln und aktuellen Nachrichten
- Übungen, einschließlich Rechnerübungen (Leistungsnachweis)

## Empfohlene Literaturliste

- M. Goodrich, et al., " Data Structures and Algorithms in Python ", John Wiley & Sons, 2013.
- R. Sedgewick and K. Wayne, " Algorithms ", Addison Wesley, 4th edition, 2011.
- M. Sipser, " Introduction to the Theory of Computation ", Cengage Learning, 3rd edition, 2012.



## SD-10 Internettechnologien

Modul Nr.	SD-10
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-10 Internettechnologien
Lehrende	Prof. Dr. Andreas Wöfl
Semester	2
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Portfolio
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

#### Fachkompetenzen

Studierende kennen Technologien, die sie bei der Gestaltung von Interaktiven Internetapplikationen nutzen können. Sie sind in der Lage diese effizient bei der Umsetzung von Projekten einzusetzen.

Die Studierenden gestalten Webseiten. Sie wissen wie man Seiten strukturiert und kennen grundlegende Sprachen um Webseiten zu gestalten (CSS, HTML, Java Script). Sie haben kleine JavaScript Programme geschrieben. Im Projekt setzten eine node.js Infrastruktur auf, integrieren einen Socketserver und realisieren Webkomponenten, um Inhalte an den Browser auszuliefern.

#### Methodenkompetenzen



Die Studierenden nutzen Kommandozeilen-Werkzeuge, um sich mit Servern zu verbinden und Daten auszutauschen. Sie nutzen Server und Client Technologien, um einfache Kommunikationen zwischen Systemen aufzubauen. Sie sind in der Lage integrierte Entwicklungsumgebungen zu nutzen.

### **Sozialkompetenzen**

Basierend auf diesen Kenntnissen führen die Studierenden ein eigenes Projekt durch. Sie wenden dabei ihr Wissen über Webtechnologien an. Sie bewerten die Ergebnisse anderer Gruppen und werden selber mit ihrem Projekt bewertet. Dabei nutzen die Studierenden Standard-Werkzeuge (GIT, Visual Code, Command Line) der Webprogrammierung.

### **Persönliche Kompetenz**

Nach Beendigung des Kurses können die Studierenden eigene Projekte durchführen und Internet (Web) Applikationen entwickeln. Im Kurs wird nicht auf Datenbanken und Netzwerktechnologien eingegangen, da diese Themen in anderen Vorlesungen verankert sind.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Dieses Modul ist Grundlage für die weiteren Informatik-Fächer und kann in anderen Studiengängen, wie Ba. Medientechnik, Ba. Interaktive Systeme oder Ba. Cyber Security verwendet werden.

## **Zugangs- bzw. empfohlene Voraussetzungen**

Empfohlen:

- Programmierung 1
- Betriebssysteme und Netzwerke

## **Inhalt**

Das Modul setzt sich aus zwei Teilen zusammen:

Teil I Internettechnologien Grundlagen und einem Teil II Projektarbeit Internettechnologien

Inhalt Teil 1

- (1) Werkzeuge und Installation
- (2) Grundlagen Client - Server, Protokolle
- (3) Client Webtechnologien
  - Html
  - CSS
  - Java Script
- (4) Server Technologien



## (5) Proprietäre Applikationen

- Sockets
- Datenformate
- Session Management

Inhalte Teil 2

Workshop: Setup Infrastruktur - Cloud based Services

Projekt: Realisierung einer Webapplikation

## Lehr- und Lernmethoden

Vorlesung, Tutorials, Praktika. Im zweiten Kursteil wird ein Projekt erarbeitet. Die Infrastruktur wird im Rahmen der Vorlesung aufgesetzt.

## Besonderes

Die Notenbildung teilt sich in Projektleistung und Prüfung. Die Projektleistung wird nach Schema bewertet. Zusätzlich gibt es eine schriftliche Prüfung, die das Grundverständnis abprüft.

## Empfohlene Literaturliste

- (1) Tutorials und Grundlagen von Internet Technologien, <https://www.w3schools.com/>
- (2) Node.js das umfassende Handbuch, Sebastian Springer, 2021, Rheinwerk Computing, ISBN 978-3-8362-8765-4
- (3) HTML5 und CSS3 für Einsteiger: Der leichte Weg zur eigenen Webseite, Paul Fuchs, 2019
- (4) JQuery 3, Frank Bongers, Rheinwerk Computing, ISBN 978-3-8362-5664-3
- (5) Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques, 3rd Edition, 2020, 978-1839211560



## SD-11 Betriebssysteme und Netzwerke 2

Modul Nr.	SD-11
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-11 Betriebssysteme und Netzwerke 2
Lehrende	Prof. Dr. Andreas Wöfl
Semester	2
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 150 Stunden Gesamt: 210 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-12 Schlüsselqualifikation 2

Modul Nr.	SD-12
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-12 Schlüsselqualifikation 2
Lehrende	Prof. Dr. Andreas Wöfl
Semester	2
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	PrA
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-13 Datenbanken

Modul Nr.	SD-13
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-13 Datenbanken
Lehrende	Prof. Dr. Andreas Wöfl
Semester	3
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Die Studierenden lernen grundlegende Konzepte von Datenbanksystemen und deren Anwendung.

Nach Abschluss des Moduls haben die Absolventen die folgenden Lernziele erreicht:

- Sie können den Entwicklungsprozess für Datenbanken beschreiben.
- Sie verstehen die grundlegenden Konzepte der DBMS-Architektur.
- Sie können Entity-Relationship-Modelle entwickeln und bewerten.
- Sie können die relationale Algebra anwenden.
- Sie können Datenbankanomalien erkennen und bewerten und normalisierte Datenbanken mit einem DBMS entwickeln.



- Sie können selbstständig SQL-Abfragen für fachspezifische Fragestellungen entwickeln.

### **Fach- und Methodenkompetenz**

Die Studierenden lernen selbstständig Datenbanken zu entwickeln. Dazu lernen die Studierenden Entity-Relationship-Modelle kennen, mit denen Datenbank konzeptioniert werden. Die Studierenden lernen des Weiteren die Entity-Relationship-Modelle in Datenbanktabellen zu überführen und dabei verschiedene Anomalien zu vermeiden. Anhand von relationaler Algebra lernen die Studierenden den grundlegenden Aufbau von Abfragesprachen zu verstehen. Mit Hilfe der Abfragesprache SQL lernen Studierende Datenbankkonzepte umzusetzen und Datenbanken zu entwickeln. Des Weiteren lernen die Studierenden selbstständig Datenbankabfragen mittels SQL zu entwickeln, mit denen verschiedene fachliche Fragen beantwortet werden können. Anhand von Normalformen lernen die Studierenden Datenbankentwürfe zu bewerten und weiterzuentwickeln.

### **Soziale Kompetenzen**

Die Studierenden lernen gemeinsam komplexe Datenbanken zu entwickeln und Datenbankentwürfe gegenseitig zu beurteilen.

### **Persönliche Kompetenz**

Die persönliche Kompetenz wird durch das strukturierte Erarbeiten von komplexen Datenbankentwürfen und komplexen Datenabfragen gefördert. Durch die theoretische Unterfütterung und praktische Anwendung von analytischen Datenbankmethoden erweitern die Studierenden insbesondere ihre Fähigkeiten im abstrakten und analytischen Denken.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Die Module Programmieren II, Programmierprojekt, Datenvisualisierung und Datenmanagement sowie Software Engineering bauen thematisch auf diesem Modul auf. Das Modul kann in anderen Studiengängen der Fakultät AI verwendet werden.

## **Zugangs- bzw. empfohlene Voraussetzungen**

empfohlen:

Modul Informatik

Die Kenntnis einer Programmiersprache ist wünschenswert.

Office-Anwendungen werden vorausgesetzt.

## **Inhalt**

- 1 Einführung
- 2 Architektur von RDBMS
- 3 Relationales Design



- 4 Relationales Modell
- 5 Datendefinition mit SQL
- 6 Datenmanipulation und -selektion mit SQL
- 7 Transaktionsmanagement

## Lehr- und Lernmethoden

- Vorlesungen
- Übungen (Learning Labs)
- Hausaufgaben

## Empfohlene Literaturliste

Thomas M. Conolly, Carolyn E. Begg: Database systems, A practical approach to design, implementation, and management. Addison-Wesley, an imprint of Pearson Education, 4th edition 2005.

Kemper A., Eickler A.: Datenbanksysteme: Eine Einführung, Oldenbourg  
Wissenschaftsverlag

Preiß, N. (2007), Entwurf und Verarbeitung relationaler Datenbanken, Oldenbourg,  
München u.a.



## SD-14 Stochastik

Modul Nr.	SD-14
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-14 Stochastik
Lehrende	Prof. Dr. Andreas Wöfl
Semester	3
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Die Studierenden haben nach Abschluss des Moduls folgende Lernziele erreicht:  
 Im Vordergrund steht die Fach- und die Methodenkompetenz in Stochastik. Die Studierenden verfügen über Kenntnisse der Konzepte der deskriptiven und induktiven Statistik. Der Erwerb von sozialen Kompetenzen steht bei diesem Modul naturgemäß nicht im Vordergrund, wird aber durch Kooperation der Studierenden und gemeinsames Erarbeiten von Lösung gefördert. Die persönliche Kompetenz wird durch vertieftes selbständiges Erarbeiten und Lösen komplexer Probleme geschärft.

Deskriptive Statistik:



Die Studierenden kennen die Konzepte der deskriptiven Statistik insbesondere für univariate und bivariate Beschreibungen. Sie sind in der Lage statistische Fragestellungen dieser Gebiete aus der betrieblichen Praxis zu erkennen, zu modellieren und zu lösen. Dazu setzen sie Softwarewerkzeuge, wie beispielsweise die Statistikfunktionen in Tabellenkalkulationsprogrammen (MS Excel, OpenOffice Calc oder LibreOffice), ein.

Induktive Statistik:

Die Studierenden kennen die Konzepte der induktiven Statistik basierend auf Wahrscheinlichkeitstheorie. Die in der Praxis vorkommenden statistischen Fragenstellung des Schließens von einer Stichprobe auf Gesamtpopulationen können je nach Themenstellung mit einer statistischen Technik des Schätzens von Parametern, dem Durchführen von parametrischen Hypothesentests und von Anpassungstests gelöst werden. Sie sind in der Lage dazu die notwendige Modellbildung mit Zufallsvariablen, Testfunktionen und ihren Wahrscheinlichkeitsverteilungen zu erstellen. Dazu setzen sie Softwarewerkzeuge, wie beispielsweise die Statistikfunktionen in Tabellenkalkulationsprogrammen (MS Excel, OpenOffice Calc oder LibreOffice), ein.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Verwendbarkeit des Moduls für Bachelor Künstliche Intelligenz:

- KI-21 Maschinelles Lernen
- KI-28 KI-Projekt
- KI-29 Deep Learning/Big Data
- KI-36 Bachelorarbeit

Verwendbarkeit des Moduls für Bachelor Cyber Security:

- CY-B-20: Wahlpflichtmodul Projekt
- CY-B-21: Kryptologie 2
- CY-B-22: Management von IT-Sicherheitsvorfällen
- CY-B-27: Digitale Forensik
- CY-B-29: Security Engineering
- CY-B-32: Auditierung von IT-Systemen

## **Zugangs- bzw. empfohlene Voraussetzungen**

empfohlen:

- Mathematik 1

## **Inhalt**

**Teil Deskriptive Statistik:**

- 1 Grundlagen und Grundbegriffe
  - Merkmale, Merkmalsträger



- Ausprägungen, Skalenniveau
- Grundgesamtheit, Voll-/Teilerhebung
- Primär- und sekundärstatistische Erhebung
- Erhebungstechniken
- 2 Häufigkeitsverteilungen
  - Urliste
  - Häufigkeitsverteilung
  - Gruppierung und Klassifikation
  - Graphischen Darstellungen
- 3 Lageparameter
  - Das arithmetische Mittel
  - Das gewogene arithmetische Mittel
  - Der Median oder Zentralwert
  - Der Modus oder Modalwert
  - Das geometrische Mittel
  - Das harmonische Mittel und das gestutzte Mittel
- 4 Streuungsmaße
  - Spannweite
  - Mittlere absolute Abweichung
  - Mittlere quadratische Abweichung (Varianz)
  - Standardabweichung
  - Quantile, Quartile und Semiquartilsabstand
  - Der Quartilkoeffizient
- 5 Konzentrationsmaße
  - absolute und relative Konzentration
  - Herfindahl-Index
  - Konzentrationsraten und Konzentrationskurven
  - Das Maß von Lorenz/Münzner
  - Der Lorenzkoeffizient
  - Die Lorenzkurve
- 6 Indexzahlen
  - Zeitreihen
  - Gliederungszahlen, Messziffern, Wachstumsraten
  - Umbasierung und Verkettung
  - Preisindex
  - Mengenindizes
  - Wertindex
- 7 Regression
  - Regressionsrechnung
  - Lineare Einfachregression
  - Die Methode der kleinsten Quadrate
  - Determinationskoeffizient



- Prognose
  - Nichtlineare Regression und Mehrfachregression
- 8 Korrelaton
- Der Korrelationskoeffizient von Bravais-Pearson
  - Eigenschaften von Varianz und Kovarianz
  - Rangkorrelation nach Spearman-Pearson
  - Korrelationsmaßzahlen für nominale Variablen

### **Teil Induktive Statistik:**

- 1 Elementare Wahrscheinlichkeitstheorie
  - Wahrscheinlichkeitsbegriffe
  - Zufallsexperimente und Ereignisse
  - Axiome nach Kolmogorov
  - Zweistufige Experimente und bedingte Wahrscheinlichkeit
  - Satz von Bayes
- 2 Zufallsvariablen
  - Zufallsvariablen
  - Diskrete Wahrscheinlichkeitsverteilungen und Verteilungsfunktion
  - Stetige Wahrscheinlichkeitsverteilungen und Dichtefunktion
  - Erwartungswert und Varianz einer Zufallsvariablen
- 3 Verteilungen I
  - Binomialverteilung
  - Normalverteilung
  - Multinomialverteilung
  - Hypergeometrische Verteilung
  - Poissonverteilung
- 4 Stichprobenverteilungen
  - Stichproben
  - Auswahlverfahren
  - Stichprobenverteilung
- 5 Zentraler Grenzwertsatz und Anwendungen
  - Zentraler Grenzwertsatz
  - Stichprobenverteilung des Mittelwerts
  - Stichprobenverteilung des Anteilswerts
  - Stichprobenverteilung der Standardabweichungen
  - Stichprobenverteilung von Differenzen
- 6 Parametrische Hypothesentests
  - Nullhypothesen und Testtheorie
  - Entscheidungsfehler
  - Tests für Mittelwert, Anteilswert, Standardabweichung und Differenzen
  - Güte eines Tests
- 7 Schätzstatistik



- Punktschätzverfahren: Momentenmethode
  - Punktschätzverfahren: Maximum-Likelihood
  - Gütekriterien
  - Intervallschätzung und Konfidenzintervall
- 8 Verteilungen II
- Student-t-Verteilung
  - Chi-Quadrat-Verteilung
  - F-Verteilung
- 9 Parametrische Hypothesentests mit kleine Stichproben
- Anteilswerttest - Binomialtest
  - Anteilswertdifferenztest - Fishertest
  - Mittelwert- und Mittelwertdifferenztest
  - Varianzquotiententest
- 10 Anpassungstests
- Verteilungshypothesen
  - Chi-Quadrat-Anpassungstest
  - Unabhängigkeitstests

## Lehr- und Lernmethoden

In klassischer Vortragstechnik werden Theorie und Anwendungen vermittelt und dargestellt. Viele Konzepte werden anhand konkreter Aufgabenstellungen erarbeitet und mit einem SW-Werkzeug gelöst. Übungsaufgaben zur eigenen Bearbeitung durch die Studierenden werden gestellt. Lösungen zu einer Auswahl davon werden zu Beginn der nächsten Vorlesung durch Studierende vorgetragen. Alternativ werden Lösungsvorschläge der Studierenden im iLearn-System diskutiert.

## Empfohlene Literaturliste

### Literatur:

- Bourier G. (2022), Beschreibende Statistik, Praxisorientierte Einführung. Mit Aufgaben und Lösungen, 14. Aufl. Gabler-Verlag, ISBN 3658370203
- Bourier G. (2018), Wahrscheinlichkeitsrechnung und schließende Statistik, Praxisorientierte Einführung. Mit Aufgaben und Lösungen, 9. akt. Aufl. Gabler-Verlag, ISBN 3658074809
- Falk, Becker, Marohn (2004), Angewandte Statistik mit SAS, Springer Verlag, Berlin
- Georgii, H.O. (2015), Stochastik, Einführung in die Wahrscheinlichkeitstheorie und Statistik, Walter de Gruyter, Berlin
- Grabmeier J., Hagl S. (2020), Statistik - Grundwissen und Formeln, 4. Auflage, Haufe Taschen Guide 215, ISBN: 978-3-648-13965-3



- Hagl, S. (2017), Crashkurs Statistik - inkl. Arbeitshilfen online. Daten erheben, analysieren und präsentieren. Haufe Verlag, ISBN: 978-3-648-09673-4
- Monka, Michael, Voss, Werner, Schöneck, Nadine (2008), Statistik am PC, Lösungen mit Excel, 5., aktualisierte und erweiterte Auflage, Hanser-Verlag, München
- Pflaumer, Heine, Hartung (2001), Statistik für Wirtschafts- und Sozialwissenschaftler, Deskriptive Statistik, Oldenbourg, München
- Puhani (2005), Statistik, Einführung mit praktischen Beispielen, Lexika-Verlag, Würzburg
- Schwarze, J. (2014), Grundlagen der Statistik: Band 1, 12. Aufl., nwb Studium.
- Schwarze, J. (2013), Grundlagen der Statistik: Band 2, 10. Aufl., nwb Studium
- Zwerenz, Karlheinz (2008), Statistik verstehen mit Excel, R. Oldenbourg Verlag, München Wien

**Internetquellen:**

- Hagl, S., VHB-Grundkurse Statistik I und II, <https://kurse.vhb.org/>



## SD-15 Projektmanagement

Modul Nr.	SD-15
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-15 Projektmanagement
Lehrende	Prof. Dr. Andreas Wöfl
Semester	3
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Portfolio
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Die Studierenden verfügen über grundlegendes, allgemeines Wissen und grundlegendes Fach- und Methodenwissen in dem Bereich Projektmanagement.

Im Einzelnen haben die Studierenden nach Abschluss des Moduls folgende Lernergebnisse erreicht:

#### Fachkompetenz

- Die Studierenden erwerben Kenntnisse im Planen, Überwachen und Steuern von Projekten und in der Gestaltung der hierfür erforderlichen Aufbau- und Ablauforganisation.

#### Methodenkompetenz

- Die Studierenden wenden ausgewählte Techniken des Projektmanagements an.



## **Persönliche Kompetenz**

- Die Studierenden erwerben Kenntnisse in der Eigenorganisation.

## **Sozialkompetenz**

- Diese Kenntnisse wenden sie in verschiedenen Teams anhand eines praxisorientierten Software- oder Organisationsprojektes an. Dadurch werden Kooperations- und Kommunikationsfähigkeit sowie Konfliktfähigkeit gefördert.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

alle Module mit umfangreicheren Gruppen-/Projektarbeiten

## **Zugangs- bzw. empfohlene Voraussetzungen**

Keine Voraussetzungen.

## **Inhalt**

- 1 Klassisches Projektmanagement
  - Erkennen der Charakteristika von Projekten im Vergleich zu Linienaufgaben in einem Unternehmen, Anforderungen an einen Projektleiter und seine Aufgaben
  - Projektorganisation - Darstellung und Diskussion unterschiedlicher Formen der Organisation eines Projektteams, Mögliche Aufgaben- und Kompetenzverteilungen zwischen Projektleiter und Linienführungskräften, Zusammensetzung, Aufgaben und Kompetenzen anderer Gremien in einer Projektorganisation
  - Projektplanung und -controlling - Darstellung unterschiedlicher Arten von Projektplänen und ihrer Abhängigkeiten, Vorgehensweise bei der Projektplanung, Darstellung des Risikomanagements in Projekten, Dimensionen der Projektsteuerung und -kontrolle mit den zugehörigen Werkzeugen, Verfahren und Vorgehensweisen
  - Projektphasen - Vorstellung ausgewählter Projektphasen, Erlernen der Aufgaben in diesen Phasen
  - Techniken - Vorstellung von Softskills eines Projektleiters (Kreativitätstechniken, Moderation, Präsentation)
  - Erwerb von Kenntnissen im Umgang mit SW zur Projektplanung und -steuerung anhand von praktischen Übungen
- 2 Agiles Projektmanagement
  - Agile Werte/Prinzipien
  - Scrum - Rollen, Ereignisse, Artefakte



- Kanban - Praktiken, Prozess, Regeln, Best Practices
- 3 Einsatzfelder und Kombination von Klassischen und Agilen Ansätzen
- 4 Projektmanagement mit MS Project
- 5 Teilweise Durchführung eines praxisorientierten Software- oder Organisationsprojektes im Team

## Lehr- und Lernmethoden

- Vorlesungen
- Übungen/Fallstudien in Einzel- und Gruppenarbeit
- Präsentationen

## Besonderes

Der Leistungsnachweis besteht aus zwei Gruppenarbeiten, die jeweils mit einer gemeinsamen 15-minütigen Präsentation abgeschlossen werden.

## Empfohlene Literaturliste

- Chatfield, C. u.a., (2011), Microsoft Project 2010 - Das offizielle Trainingsbuch, O`Reilly, Köln
- GPM Deutsche Gesellschaft für Projektmanagement, Gessler, M. (Hrsg.) (2019), Kompetenzbasiertes Projektmanagement (PM4)- Handbuch für die Projektarbeit, Qualifizierung und Zertifizierung auf Basis der IPMA Competence Baseline Version 4, 1. Auflage, GPM Deutsche Gesellschaft für Projektmanagement, Nürnberg
- Kerzner, H. (2003), Projektmanagement Fallstudien, mitp-Verlag, Bonn
- Kuster, J. et al. (2019), Handbuch Projektmanagement, 4. Auflage, Springer Verlag, Berlin
- Martinelli, R.J., Milosevic, D.Z. (2016), Project Management ToolBox - Tools and Techniques for the Practicing Project Manager, 2. Auflage, Wiley, Hoboken, NJ
- Project Management Institute (Hrsg.) (2017), A guide to the project management body of knowledge. PMBOK(R) Guide, 6. Auflage, Project Management Institute, Newtown Square, Pa
- Schwaber, K., Sutherland, J. (2016), Der Scrum Guide, Scrum.Org and ScrumInc, o.O.
- Timinger, H. (2017), Modernes Projektmanagement: Mit traditionellem, agilem und hybridem Vorgehen zum Erfolg, Wiley, Hoboken, NJ
- Verzuh, E. (2016), The Fast Forward MBA in Project Management, 5. Auflage, Wiley, Hoboken, NJ
- Wies, P. (2014), Project 2013 Grundlagen, Herdt-Verlag, Bodenheim



## SD-16 Moderne Programmierparadigmen

Modul Nr.	SD-16
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-16 Moderne Programmierparadigmen
Lehrende	Prof. Dr. Andreas Wöfl
Semester	3
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Portfolio
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-17 HCI (Human Computer Interaction)

Modul Nr.	SD-17
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-17 HCI (Human Computer Interaction)
Lehrende	Prof. Dr. Andreas Wöfl
Semester	3
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-18 Schlüsselqualifikation 3

Modul Nr.	SD-18
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-18 Schlüsselqualifikation 3
Lehrende	Prof. Dr. Andreas Wöfl
Semester	3
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-19 Software Engineering

Modul Nr.	SD-19
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-19 Software Engineering
Lehrende	Prof. Dr. Andreas Wöfl
Semester	4
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Portfolio
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Die Studierenden verfügen über detailliertes Fachwissen und Methodenwissen im Bereich der Softwareentwicklung.

Im Einzelnen haben die Studierenden nach Abschluss des Moduls folgende Lernergebnisse erreicht:

#### Fachkompetenz

- Die Studierenden können die Grundlagen des Projektmanagements anwenden.
- Sie können Anforderungen eines Software-Projekts verstehen und bewerten.
- Sie kennen die Codierregeln und können diese anwenden.



- Sie können Software mittels CI/CD automatisiert bauen, testen, und pakettisieren.
- Sie sind in der Lage Reviews von Arbeitsergebnissen durchzuführen.

### **Methodenkompetenz**

- Sie sind in der Lage aus Anforderungen auf systematische Weise einen objektorientierten Entwurf (Analyse und Design) mittels UML anzufertigen und in Code zu überführen.
- Sie können ausgehend von Anforderungen und auf Basis des Codes Testfälle gemäß Black-Box- und White-Box-Teststrategien definieren, Testenkriterien festlegen und Tests durchführen.
- Sie kennen die Komponenten des CI/CD Ansatzes und können eigenständig aus den Anforderungen Pipelines ableiten.

### **Persönliche Kompetenz**

- Durch zielorientiertes Arbeiten entwickeln die Studierenden ein hohes Maß an Zielstrebigkeit.
- Durch agile Methoden wird die Selbstmotivation der Studierenden gefördert.
- Durch die Task-orientierte Arbeitsweise wird das problemlösende Denken der Studierenden geschärft.

### **Sozialkompetenz**

- Die Studierenden sind in der Lage sich selbständig für ein Projekt in Arbeitsgruppen zu organisieren und das Projekt gemeinsam durchzuführen.
- Durch die aktive Teilnahme an Teammeetings wird die Teamfähigkeit gestärkt.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Wahlpflichtmodul anderer Bachelorstudiengänge (wie z.B.: Angewandte Informatik/Infotronik, Interaktive Systeme/Internet of Things, Künstliche Intelligenz, Wirtschaftsinformatik, Elektro- und Informationstechnik)

## **Zugangs- bzw. empfohlene Voraussetzungen**

### **Zugangsvoraussetzungen:**

- keine spezifischen

### **empfohlene Voraussetzungen:**

- Kenntnisse der Inhalte der Module
  - Grundlagen der Informatik
  - Programmierung 1
  - Programmierung 2
  - Sichere Programmierung (Bachelor Cyber Security)



## Inhalt

- 1 Motivation und Definition
- 2 Elemente des Software Engineering
- 3 Methodik
  - Requirements Engineering
  - Software Entwurf (allgemein)
  - Software Entwurf
  - Architektur und Detaildesign allgemein
  - Objektorientierte Analyse und Design (OOA, OOD)
  - UML Einführung
  - UML Workshop (Diagramme und ihre Anwendung)
  - Anwendungsbeispiel
  - Übergang von Analyse zum Design
- 4 Implementierung
  - Codierungsregeln (z.B. MISRA)
  - Statische Codeanalyse
  - Codemetriken
- 5 Software Test
  - Statischer Test
  - Dynamischer Test
  - Testprozeß
  - Testmethoden und Teststrategien
- 6 Software Qualitätssicherung
  - Definition
  - Reviews

## Lehr- und Lernmethoden

- Seminaristischer Unterricht mit praktischen Übungen, teilweise Gruppenarbeit
- Semesterbegleitende Projektarbeit in Gruppenarbeit

## Empfohlene Literaturliste

- H. Balzer, Lehrbuch der Software-Technik, Spektrum Akademischer Verlag
- I. Sommerville, Software Engineering, Addison Wesley Verlag
- B. Kahlbrandt, Software-Engineering mit der UML, Springer Verlag
- C Rupp et. al., UML 2 - Glasklar, Hanser Verlag
- A. Spillner, T. Linz, Basiswissen Softwaretest, dpunkt Verlag
- B. Beizer, Black - Box Testing: Techniques for Functional Testing of Software and Systems, Wiley Verlag



- P. Liggesmeyer, Software - Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum Verlag
- H. Sneed, M. Winter, Testen objektorientierter Software, Hanser Verlag



## SD-20 Datenbanken 2

Modul Nr.	SD-20
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-20 Datenbanken 2
Lehrende	Prof. Dr. Andreas Wöfl
Semester	4
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Undergraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Portfolio
Gewichtung der Note	5/210
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Die Studierenden lernen grundlegende Konzepte von Datenbanksystemen und deren Anwendung.

Nach Abschluss des Moduls haben die Absolventen die folgenden Lernziele erreicht:

- Sie können den Entwicklungsprozess für Datenbanken beschreiben.
- Sie verstehen die grundlegenden Konzepte der DBMS-Architektur.
- Sie können Entity-Relationship-Modelle entwickeln und bewerten.
- Sie können die relationale Algebra anwenden.
- Sie können Datenbankanomalien erkennen und bewerten und normalisierte Datenbanken mit einem DBMS entwickeln.
- Sie können selbstständig SQL-Abfragen für fachspezifische Fragestellungen entwickeln.



### **Fach- und Methodenkompetenz**

Die Studierenden lernen selbstständig Datenbanken zu entwickeln. Dazu lernen die Studierenden Entity-Relationship-Modelle kennen, mit denen Datenbank konzeptioniert werden. Die Studierenden lernen des Weiteren die Entity-Relationship-Modelle in Datenbanktabellen zu überführen und dabei verschiedene Anomalien zu vermeiden. Anhand von relationaler Algebra lernen die Studierenden den grundlegenden Aufbau von Abfragesprachen zu verstehen. Mit Hilfe der Abfragesprache SQL lernen Studierende Datenbankkonzepte umzusetzen und Datenbanken zu entwickeln. Des Weiteren lernen die Studierenden selbstständig Datenbankabfragen mittels SQL zu entwickeln, mit denen verschiedene fachliche Fragen beantwortet werden können. Anhand von Normalformen lernen die Studierenden Datenbankentwürfe zu bewerten und weiterzuentwickeln.

### **Soziale Kompetenzen**

Die Studierenden lernen gemeinsam komplexe Datenbanken zu entwickeln und Datenbankentwürfe gegenseitig zu beurteilen.

### **Persönliche Kompetenz**

Die persönliche Kompetenz wird durch das strukturierte Erarbeiten von komplexen Datenbankentwürfen und komplexen Datenabfragen gefördert. Durch die theoretische Unterfütterung und praktische Anwendung von analytischen Datenbankmethoden erweitern die Studierenden insbesondere ihre Fähigkeiten im abstrakten und analytischen Denken.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Die Module Programmieren II, Programmierprojekt, Datenvisualisierung und Datenmanagement sowie Software Engineering bauen thematisch auf diesem Modul auf. Das Modul kann in anderen Studiengängen der Fakultät AI verwendet werden.

## **Zugangs- bzw. empfohlene Voraussetzungen**

formal: keine

inhaltlich: Modul Grundlagen der Informatik

## **Inhalt**

- 1 Einführung
- 2 Architektur von RDBMS
- 3 Relationales Design
- 4 Relationales Modell
- 5 Datendefinition mit SQL
- 6 Datenmanipulation und -selektion mit SQL
- 7 Transaktionsmanagement



## Lehr- und Lernmethoden

- Vorlesungen
- Übungen (Learning Labs)
- Hausaufgaben

## Empfohlene Literaturliste

Thomas M. Conolly, Carolyn E. Begg: Database systems, A practical approach to design, implementation, and management. Addison-Wesley, an imprint of Pearson Education, 4th edition 2005.

Kemper A., Eickler A.: Datenbanksysteme: Eine Einführung, Oldenbourg  
Wissenschaftsverlag

Preiß, N. (2007), Entwurf und Verarbeitung relationaler Datenbanken, Oldenbourg,  
München u.a.



## SD-21 Cloud Computing

Modul Nr.	SD-21
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-21 Cloud Computing
Lehrende	Prof. Dr. Andreas Wöfl
Semester	4
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
Niveau	Postgraduate
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Portfolio
Gewichtung der Note	5/90
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

Erwerb der Kenntnis und Fähigkeit cloudbasierte Infrastruktur zu planen, implementieren und verwalten sowie Cloud-Dienste effektiv und sicher zu nutzen. Im Einzelnen haben die Studierenden nach Abschluss des Moduls folgende Lernziele erreicht:

#### Fachkompetenz:

- Die Studierenden erlernen die Fertigkeiten zum Entwurf von cloud-native Software mit Fokus auf Microservices.
- Die Studierenden sind in der Lage, Anwendungen mithilfe von Docker zu containerisieren.
- Die Studierenden können containerisierte Anwendungen auf der Kubernetes-Plattform orchestrieren, skalieren und überwachen.



- Die Studierenden erlernen die Fertigkeit zum Entwurf und zur Implementierung serverloser Anwendungen.
- Die Studierenden sind in der Lage, cloud-basierte Infrastrukturen wie virtuelle Maschinen oder private Netzwerke mithilfe moderner Tools automatisiert bereitzustellen.
- Die Studierenden können ein privates on-premise Netzwerk mit einem privaten Cloud-Netzwerk verbinden und es hinsichtlich IT-Sicherheit konfigurieren.
- Die Studierenden können ein Konzept für Identitäts- und Zugriffsmanagement spezifisch eine Organisation erstellen und technisch umsetzen.

### **Methodenkompetenz:**

- Die Studierenden erwerben Kenntnisse über die theoretischen Grundlagen von verteilten Systemen.
- Die Studierenden lernen verschiedene Betriebsformen wie Infrastructure-as-a-Service und Software-as-a-Service kennen und können diese je nach Anwendungsfall bewerten und einordnen.
- Die Studierenden erlangen Kenntnisse über grundlegende Konzepte und Technologien, die für den strukturierten Aufbau cloudbasierter Infrastrukturen notwendig sind.
- Die Studierenden erlangen Kenntnisse über die wichtigsten Entwurfsmuster cloud-nativer Software.
- Die Studierenden lernen die Grundlagen und Herausforderungen von Serverless-Architekturen kennen und verstehen deren Auswirkungen auf die Gesamtarchitektur von Softwaresystemen.
- Die Studierenden erlangen Kenntnisse über den Infrastructure-As-Code-Ansatz und können die Konzepte und Technologien den angebotenen Diensten öffentlicher Cloud-Provider zuordnen.
- Die Studierenden lernen Automatisierungsmethoden wie GitOps kennen, um cloudbasierte Infrastrukturen und cloud-native Software sicher und zuverlässig bereitzustellen.
- Die Studierenden erwerben Kenntnisse und Fertigkeiten zur Verwaltung von Identitäts- und Zugriffsmanagement anhand von öffentlichen Cloud-Providern.
- Die Studierenden lernen die grundlegenden Konzepte der Cloud-Sicherheit mit Fokus auf Netzwerksicherheit cloudbasierter Systeme.
- Die Studierenden erlangen Kenntnisse über die datenschutzrechtlichen Vorschriften und Best Practices bei der Datenverarbeitung in der Cloud.

### **Persönliche Kompetenz:**

- Die Studierenden können in praxisnahen und praktischen Laborübungen eigene Ideen ausarbeiten, verfeinern, analysieren und testen.



- Die Studierenden lernen in Teams zu arbeiten, ihre Ideen zu kommunizieren und Ihre Aufgaben zu koordinieren.
- Die Studierenden werden ermutigt, kreativ zu denken und neue Ansätze zur Nutzung der Cloud-Technologien zu entwickeln.
- Die Studierenden sind in der Lage Systemdenken anzunehmen und Schlüsselprobleme zu identifizieren.
- Die Studierenden lernen ihre Zeit effektiv zu organisieren, Prioritäten zu setzen und ihre Aufgaben innerhalb der vorgegebenen Zeiträume zu erledigen.
- Die Studierenden erlernen die Fähigkeit fundierte Empfehlungen zu Themen des Cloud Computing abzugeben.

## **Verwendbarkeit in diesem und in anderen Studiengängen**

Pflichtfach im Master-Studiengang Wirtschaftsinformatik, verwendbar in thematisch verwandten Studiengängen und Fächern.

## **Zugangs- bzw. empfohlene Voraussetzungen**

Formal: Keine

Inhaltlich:

- Betriebssysteme
- Netzwerke
- Programmierung I + II
- Software Engineering

(empfohlene Voraussetzung)

## **Inhalt**

- 1 Einführung in Cloud Computing
- 2 Fundamentale Konzepte und Betriebsmodelle
- 3 Microservice Architektur
- 4 Entwurfsmuster Cloud-Nativer Software
- 5 Container-Basierte Virtualisierung
- 6 Container Orchestrierung
- 7 Serverless Architektur
- 8 Infrastructure-As-Code
- 9 Virtuelle Netzwerke
- 10 Cloud Deployment und Management
- 11 Cloud-Sicherheit und Datenschutz
- 12 Identitäts- und Zugriffsmanagement



## Lehr- und Lernmethoden

- Seminaristischer Unterricht mit praktischen Übungen
- Semesterübergreifende Laborübungen

## Empfohlene Literaturliste

Thomas Erl, Zaigham Mahmood, Ricardo Puttini, *Cloud Computing ? Concepts, Technology & Architecture* , 2nd ed., Prentice Hall, 2023

Douglas E. Comer, *The Cloud Computing Book: The Future of Computing Explained* , 1st ed., CRC Press, 2021

Cornelia Davis, *Cloud Native Patterns* , 1st ed., Manning Publications Co., 2019

Jeff Nickoloff, Stephen Kuenzli, *Docker in Action* , 2nd ed., Manning Publications Co., 2019

Brendan Burs, Joe Beda, Kelsey Hightower, Lachlan Evenson, *Kubernetes Up & Running* , 3rd ed., O`Reilly Media, 2022

Kief Morris, *Infrastructure As Code: Managing Servers in the Cloud* , 1st ed., O`Reilly Media, 2015

Jeff Geerling, *Ansible for DevOps: Server and Configuration Management for Humans* , 1st ed., Leanpub, 2020

Nicole Forsgren, Jez Humble, Gene Kil: *Accelerate: Building and Scaling High Performing Technology Organizations* , 1st ed., IT Revolution, 2018

Stian Thorgersen, *Keycloak ? Identity and Access Mangement for Modern Applications* , 1st ed., Packt Publishing, 2021



## SD-22 IT Sicherheit

Modul Nr.	SD-22
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-22 IT Sicherheit
Lehrende	Prof. Dr. Andreas Wöfl
Semester	4
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	PrA
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-23 FWP 1 (aus Softwaretechnologie)

Modul Nr.	SD-23
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-23 FWP 1 (aus Softwaretechnologie)
Lehrende	Prof. Dr. Andreas Wöfl
Semester	4
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Prüfungsart des gewählten Moduls
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-24 Schlüsselqualifikation 4

Modul Nr.	SD-24
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-24 Schlüsselqualifikation 4
Lehrende	Prof. Dr. Andreas Wöfl
Semester	4
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	schr. P. 90 Min.
Dauer der Modulprüfung	90 Min.
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-25 Praxismodul

Modul Nr.	SD-25
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-25 Praxisergänzende Vertiefung 1 SD-5101 Betriebspraktikum SD-5103 Praxisergänzende Vertiefung 2
Lehrende	Prof. Dr. Andreas Wöfl
Semester	5
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	30
Workload	Präsenzzeit: 0 Stunden Gesamt: 0 Stunden
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-26 SE Projekt 1

Modul Nr.	SD-26
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-26 SE Projekt 1
Lehrende	Prof. Dr. Andreas Wöfl
Semester	6
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	PrA
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-27 Informatik-Seminar

Modul Nr.	SD-27
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-27 Informatik-Seminar
Lehrende	Prof. Dr. Andreas Wöfl
Semester	6
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Portfolio
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-28 Systems Engineering

Modul Nr.	SD-28
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-28 Systems Engineering
Lehrende	Prof. Dr. Andreas Wöfl
Semester	6
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	PrA
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-29 Software Engineering 2

Modul Nr.	SD-29
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-29 Software Engineering 2
Lehrende	Prof. Dr. Andreas Wöfl
Semester	6
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Portfolio
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-30 FWP 2

Modul Nr.	SD-30
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-30 FWP 2
Lehrende	Prof. Dr. Andreas Wöfl
Semester	6
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Prüfungsart des gewählten Moduls
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-31 Schlüsselqualifikation 5

Modul Nr.	SD-31
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-31 Schlüsselqualifikation 5
Lehrende	Prof. Dr. Andreas Wöfl
Semester	6
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Portfolio
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-32 SE Projekt 2

Modul Nr.	SD-32
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-32 SE Projekt 2
Lehrende	Prof. Dr. Andreas Wöfl
Semester	7
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	PrA
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## SD-33 FWP 3

Modul Nr.	SD-33
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-33 FWP 3
Lehrende	Prof. Dr. Andreas Wöfl
Semester	7
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Prüfungsart des gewählten Moduls
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls



## **SD-34 Wahlpflichtmodul 4 Unternehmensgründung / Compliance, Datenschutz**

Modul Nr.	SD-34
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-34 Wahlpflichtmodul 4 Unternehmensgründung / Compliance, Datenschutz
Lehrende	Prof. Dr. Andreas Wöfl
Semester	7
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	4
ECTS	5
Workload	Präsenzzeit: 60 Stunden Selbststudium: 90 Stunden Gesamt: 150 Stunden
Prüfungsarten	Prüfungsart des gewählten Moduls
Unterrichts-/Lehrsprache	Deutsch

### **Qualifikationsziele des Moduls**



## SD-35 Bachelormodul

Modul Nr.	SD-35
Modulverantwortliche/r	Prof. Dr. Andreas Wöfl
Kursnummer und Kursname	SD-7101 Bachelorarbeit SD-7102 Bachelorseminar
Lehrende	Prof. Dr. Andreas Wöfl
Semester	7
Dauer des Moduls	1 Semester
Häufigkeit des Moduls	jährlich
Art der Lehrveranstaltungen	Pflichtfach
SWS	2
ECTS	5
Workload	Präsenzzeit: 0 Stunden Gesamt: 0 Stunden
Unterrichts-/Lehrsprache	Deutsch

### Qualifikationsziele des Moduls

