

Fakultät Informationstechnik

Modulhandbuch SPO7

Bachelor-Studiengang

Softwaretechnik und Medieninformatik (SWB)

- Studienschwerpunkt Softwaretechnik (SWB-SWT)

Hinweise:

Die in den Modulbeschreibungen genannten Voraussetzungen sind nicht zwingend, aber sehr hilfreich für das Verständnis der vermittelten Lerninhalte.

Abkürzungen:

SWS Semesterwochenstunden

ECTS European Credit Transfer and Accumulation System

Europäisches System zur Übertragung und Akkumulierung von Studienleistungen

ECTS ist ein Maß für den erforderlichen Arbeitsaufwand im Studium (Workload)

1 ECTS entspricht näherungsweise 30 Arbeitsstunden

Die Angabe der ECTS-Punkte in den Modulbeschreibungen soll den aufzubringenden Workload transparent machen.

Version: 01.09.2019

Inhaltsverzeichnis

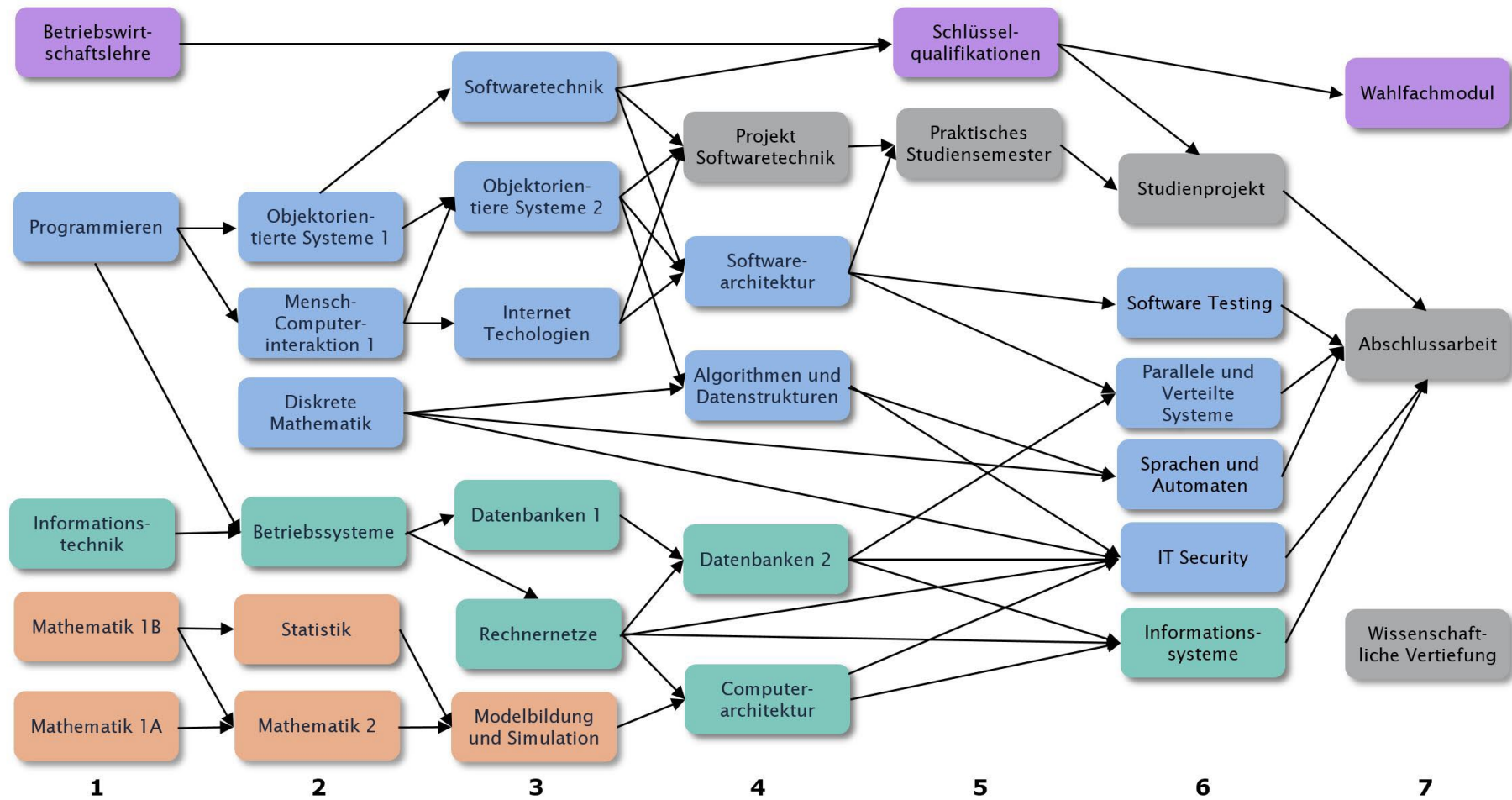
Semester	Modul	Nummer	Seite
	Übersicht Modulplan		1
1. Semester			
	Betriebswirtschaftslehre	IT 105 2033	3
	Informationstechnik	IT 105 1002	5
	Mathematik 1A	IT 105 1003	7
	Mathematik 1B	IT 105 1004	9
	Programmieren	IT 105 1015	11
2. Semester			
	Betriebssysteme	IT 105 2004	13
	Diskrete Mathematik	SWB 105 2024	15
	Mathematik 2	SWB 105 2003	17
	Mensch-Computer-Interaktion 1	IT 105 2001	19
	Objektorientierte Systeme 1	IT 105 2028	21
	Statistik	IT 105 2018	23
3. Semester			
	Datenbanken 1	IT 105 3007	25
	Internet Technologien	IT 105 3010	27
	Modellbildung und Simulation	SWB 105 3003	29
	Objektorientierte Systeme 2	SWB 105 3029	31
	Rechnernetze	IT 105 3008	33
	Softwaretechnik	IT 105 3039	35
4. Semester			
	Algorithmen und Datenstrukturen	SWB 105 3012	37
	Computerarchitektur	IT 105 4003	39
	Datenbanken 2	IT 105 4018	41
	Projekt Softwaretechnik	SWB 105 4026	43
	Softwarearchitektur	IT 105 4007	45
5. Semester			
	Praktisches Studiensemester	IT 105 5000	47
	Schlüsselqualifikationen	IT 105 5001	48
6. Semester			
	Informationssysteme	IT 105 6001	50
	IT Security	SWB 105 6005	52
	Parallel und Verteilte Systeme	SWB 105 6039	54
	Software Testing	SWB 105 6043	56
	Sprachen und Automaten	SWB 105 6042	58
	Studienprojekt	IT 105 6007	60

Semester	Modul	Nummer	Seite
7. Semester			
	Bachelorarbeit	IT 105 7000	62
	Wahlfachmodul	MD 7630	64
	Wissenschaftliche Vertiefung	IT 105 7001	66

Übersicht Modulplan Studienschwerpunkt Softwaretechnik



Übersicht Modulabhängigkeiten Studienschwerpunkt Softwaretechnik - Erreichen des Gesamtziels



Legende: Mathematik IT-Systeme Informatik / Softwaretechnik Schlüsselkompetenzen Wissenschaftliches Arbeiten

Hinweis: Die Pfeile stellen die Modulverbindung dar, die zum Erreichen des Gesamtziels beitragen. Verbindungen zwischen Modulen innerhalb eines Semesters wurden zugunsten der Übersichtlichkeit nicht dargestellt.

Modulbeschreibung Betriebswirtschaftslehre

Schlüsselworte: Betriebswirtschaftslehre, Volkswirtschaftslehre,
Mikroökonomie, Makroökonomie

Zielgruppe:	1. Semester SWB	Modulnummer:	IT 105 2033
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Dirk Hesse		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

keine

Modulziel – angestrebte Lernergebnisse:

Die Studierenden überblicken die unterschiedlichen Teilbereiche der allgemeinen Betriebswirtschaftslehre und können deren grundlegenden Instrumente und Methoden anwenden. Sie sind zudem in der Lage, mikro- und makroökonomische Aspekte unternehmerischen Handelns nachzuvollziehen und zu beschreiben. Sie kennen die Funktionsweisen und Zusammenhänge betrieblicher Strukturen und Prozesse. Sie verstehen die Notwendigkeit des Wirtschaftens als Basis für unternehmerische Vorgehensweisen und Techniken und sind in der Lage, grundlegende Methoden und Instrumente der Betriebswirtschaftslehre in ihrer Wirkung einzuschätzen und anzuwenden. Die Studierenden verstehen die prinzipielle Funktionsweise von Märkten und können grundlegende Methoden der Volkswirtschaftslehre auf einzel- und gesamtwirtschaftliche Fragestellungen anwenden. Sie verstehen die makroökonomischen Zusammenhänge von Güter-, Arbeits- und Geldmarkt.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Unternehmensformen
- betriebliche Funktionsbereiche
- Wachstum und Konjunktur

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- grundlegende Methoden und Instrumente der Betriebswirtschaftslehre in ihrer Wirkung einzuschätzen

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- sich in Projektteams zu integrieren und verantwortungsbewusst zu handeln.

Inhalt:

- Unternehmen, Rechtsformen, Typologie, Umfeld
- Aufgaben, Maßnahmen und Methoden der betrieblichen Funktionsbereiche
- Betriebliche Leistungs- und Finanzprozesse
- Grundlagen des Rechnungswesens
- Funktionsweise von Märkten, Preisbildung
- Rolle der Unternehmen und des Staats in der Marktwirtschaft
- Wachstum und Konjunktur
- Geld- und Finanzsysteme

Literaturhinweise:

- Schierenbeck; Grundzüge der Betriebswirtschaftslehre; Oldenbourg Verlag, 2012, ISBN 9783486273229.
- Vahs, Schäfer-Kunz; Schäffer-Poeschel, Einführung in die Betriebswirtschaftslehre; Schäffer-Poeschel Verlag Stuttgart, 2015, ISBN 978-3-7910-3456-0.
- Bofinger, Mayer; Grundzüge der Volkswirtschaftslehre; Pearson Studium, 2011, ISBN 978-3-86894-230-3.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Informationstechnik

Schlüsselwörter: Methodische Anwendung eines Rechners

Zielgruppe:	1. Semester SWB	Modulnummer:	IT 105 1002
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		75 h
	Selbststudium		45 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Reiner Marchthaler		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

keine

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben ein grundlegendes Verständnis für die Arbeitsweise eines Computers. Sie haben Grundkenntnisse über den grundlegenden Aufbau, die Architektur und die prinzipielle Funktionsweise eines modernen Rechners. Darüber hinaus ist ein Grundverständnis für die Codierung von Zahlen und Zeichen sowie für kombinatorische Logik vorhanden. Sie sind in der Lage, die Besonderheiten verschiedener Betriebssysteme darlegen zu können.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Arbeitsweise eines Computers,
- die Architektur moderner Rechner,
- die Zahlendarstellung in Computern.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- Boolesche Algebra anwenden und einfach kombinatorische Schaltungen entwickeln.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- verschiedene Rechnerarchitekturen und die Besonderheiten verschiedener Betriebssysteme zu benennen.

Inhalt:

- Aufgaben und Einsatzgebiete von Rechnern
- Zahlen- und Zeichencodierung (Zahlenbereich, Auflösung, Überläufe)
- Boolesche Algebra und Kombinatorische Schaltungen
- Aufbau und Architektur eines modernen Rechners
- Aufbau einer CPU, Speicher und Ein-/Ausgabe
- Überblick Betriebssysteme und Anwendungsprogramme

Literaturhinweise:

- Gumm, Heinz-Peter und Sommer, Manfred: Einführung in die Informatik, 10. Auflage, Oldenbourg Verlag, 2013.
- Hoffmann, Dirk: Grundlagen der Technischen Informatik, Hanser Verlag, 2013.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Mathematik 1A

Schlüsselworte: Funktionen, Differential- und Integralrechnung

Zielgruppe: 1. Semester SWB **Modulnummer:** IT 105 1003

Arbeitsaufwand: 5 ECTS **150 h**
Davon
Kontaktzeit **75 h**
Selbststudium **45 h**
Prüfungsvorbereitung **30 h**

Unterrichtssprache: Deutsch
Modulverantwortung: Prof. Dr. Jürgen Koch

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Elementarmathematik aus der Schule, insbesondere Kenntnisse über Funktionen

Modulziel – angestrebte Lernergebnisse:

- Die Studierenden werden in die Lage versetzt, mathematische Problemstellungen mit Funktionen analytisch zu lösen

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Eigenschaften von Funktionen in einer und in mehreren Veränderlichen
- anschauliche und mathematische Bedeutung der Begriffe Grenzwert, Stetigkeit, Ableitung und Integral

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- mithilfe von Differential- und Integralrechnung Eigenschaften von Funktionen analytisch zu bestimmen

Übergreifende Kompetenzen

Die Studierenden können

- Problemstellungen systematisch zu analysieren und zu lösen
- logische Schlussfolgerungen nachvollziehen

Inhalt:

- Elementare Funktionen und ihre Eigenschaften
- Folgen, Grenzwerte und Stetigkeit
- Differentialrechnung
- Integralrechnung
- Funktionen mit mehreren Variablen

Literaturhinweise:

- J. Koch, M. Stämpfle: Mathematik für das Ingenieurstudium, 4. Auflage, Hanser Verlag 2017, ISBN 9783446451667

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur 90 Minuten
Anteil Semesterwochenstunden:	5 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Mathematik 1B

Schlüsselworte: Vektoren, Matrizen, komplexe Zahlen

Zielgruppe: 1. Semester WKB **Modulnummer:** IT 105 1004

Arbeitsaufwand: 5 ECTS **150 h**
Davon
Kontaktzeit **75 h**
Selbststudium **45 h**
Prüfungsvorbereitung **30 h**

Unterrichtssprache: Deutsch
Modulverantwortung: Prof. Dr. Jürgen Koch

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Elementarmathematik aus der Schule

Modulziel – angestrebte Lernergebnisse:

- Die Studierenden werden in die Lage versetzt, mathematische Problemstellungen mit Vektoren, Matrizen und komplexen Zahlen analytisch zu lösen

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Begriffe und Eigenschaften von Vektoren, Matrizen und komplexen Zahlen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Berechnungen mit linearen Gleichungssystemen, Vektoren, Matrizen und komplexen Zahlen durchzuführen

Übergreifende Kompetenzen

Die Studierenden können

- Problemstellungen systematisch zu analysieren und zu lösen
- logische Schlussfolgerungen nachvollziehen

Inhalt:

- Lineare Gleichungssysteme
- Vektoren
- Matrizen
- komplexe Zahlen

Literaturhinweise:

- J. Koch, M. Stämpfle: Mathematik für das Ingenieurstudium, 4. Auflage, Hanser Verlag 2017, ISBN 9783446451667

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	5 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Programmieren

Schlüsselworte: Programmierkonzepte, Algorithmen

Zielgruppe: 1. Semester SWB **Modulnummer:** IT 105 1015

Arbeitsaufwand: 10 ECTS **300 h**
Davon **Kontaktzeit** **240 h**
Selbststudium **30 h**
Prüfungsvorbereitung **30 h**

Unterrichtssprache: Deutsch
Modulverantwortung: Prof. Dr. Mirco Sonntag

Stand: 01.09.2019

Empfohlene Voraussetzungen:

keine

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, technische Aufgabenstellungen zu verstehen, einen Algorithmus zur Lösung der Aufgabe zu entwickeln und anschließend auf Basis des Algorithmus ein Programm in einer Programmiersprache zu erstellen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- atomare Befehle und Kontrollstrukturen einer Programmiersprache
- Variablen und Konstanten
- elementare, abgeleitete und zusammengesetzte Datentypen
- das Prinzip der prozeduralen Programmierung
- ein Werkzeug zur Erstellung von Programmen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- von Aufgabenstellungen Algorithmen abzuleiten
- aus diesen Algorithmen selbstständig Programme zu entwickeln
- grundlegende Entscheidungen über den Programmentwurf zu treffen

Übergreifende Kompetenzen

Die Studierenden können

- mit einer integrierten Entwicklungsumgebung Programme erstellen

Inhalt:

- Grundlagen
 - Programmieren
 - Werkzeuge der Programmerstellung
 - Umsetzung von Aufgabenstellungen in Algorithmen
 - Speicherverwaltung, Stack und Heap
- Einführung in eine Programmiersprache
 - Elementare Datentypen, Variablen und Konstanten
 - Abgeleitete und zusammengesetzte Datentypen (Felder, Zeichenketten, Strukturen, Zeiger)
 - Ausdrücke mit Operatoren und Zuweisungen
 - Kontrollstrukturen zur Verzweigung und Iteration
 - Prozedurale Programmierung, call-by-value und call-by-reference
 - Rekursive Funktionen
 - Operationen auf Dateien

Literaturhinweise:

- Dausmann et al., C als erste Programmiersprache. Vieweg+Teubner, 2010.
- Erenkötter: C von Anfang an. rororo, 1999.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lernergebnisse:

Die Studierenden beherrschen die Methoden zur Erstellung von Programmen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Betriebssysteme

Schlüsselwörter: Prozess-/ Speicherverwaltung, IPC, Systemprogrammierung, UNIX

Zielgruppe: 2. Semester SWB **Modulnummer:** IT 105 2004

Arbeitsaufwand: 5 ECTS **150 h**
Davon **Kontaktzeit** 75 h
Selbststudium 45 h
Prüfungsvorbereitung 30 h

Unterrichtssprache: Deutsch
Modulverantwortung: Prof. Dr.-Ing. Rainer Keller

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Kenntnisse im Programmieren mit C

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben die Kompetenz zur Nutzung von Computer-Hardware und Software sowie von Betriebssystemen und Rechnernetzen. Sie können die grundlegenden Konzepte von Betriebssystemen beschreiben und die in den marktgängigen Betriebssystemen realisierten Lösungen bewerten.

Kenntnisse – fachliche Kompetenzen

Sie kennen

- die wesentlichen Funktionen und Dienste von Betriebssystemen und sind in der Lage, sie interaktiv oder in Anwendungsprogrammen zu nutzen,
- die Mechanismen der Authentisierung und Autorisierung.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind können

- den Zugriff von Nutzern auf Computer, Dienste und Daten angemessen zu regeln.

Übergreifende Kompetenzen

Die Studierenden in der Lage,

- die grundlegenden Konzepte von Betriebssystemen beschreiben und die in den marktgängigen Betriebssystemen realisierten Lösungen bewerten.

Inhalt:

- Einführung in die Aufgaben und die Struktur von Betriebssystemen
- Benutzung von UNIX und Windows per Kommandozeile (Shell- / Skript-Programmierung)
- Prozesse und Threads
- Linux Kernel Module
- Speicherverwaltung
- Interprozesskommunikation und Synchronisation
- Dateisysteme
- Input und Output
- Security
- Container, Virtualisierung und Cloud

Literaturhinweise:

A.S. Tanenbaum: Moderne Betriebssysteme, 4. Akt. Auflage, Pearson 2016

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Nachbereitung und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden sind in der Lage, ein vernetztes UNIX-System sowohl von der Kommandozeile als auch von einer grafischen Benutzungsoberfläche aus zu bedienen und häufig wiederkehrende Aufgaben durch Shell-Skripte zu automatisieren. Sie beherrschen die Programmierung von Anwendungen, die die Funktionen und Dienste des Betriebssystems durch POSIX-konforme Programmierschnittstellen nutzen. Die Studierenden sind befähigt, die wichtigsten Netzwerkdienste von Betriebssystemen Client-seitig nutzen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Diskrete Mathematik

Schlüsselworte: Zahlentheorie, Algebra,

Zielgruppe:	2. Semester SWB	Modulnummer:	SWB 105 2024
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Karin Melzer		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Lineare Gleichungssysteme, Vektoren, Matrizen, Funktionen in einer und in mehreren reellen Veränderlichen, komplexe Zahlen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden können konkrete Anwendungen in der Informatik durch abstrakte mathematische Methoden analysieren und lösen. Sie werden in die Lage versetzt, mathematische Problemstellungen der Theoretischen Informatik und der Kryptografie mathematisch zu lösen

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Beweistechniken
- Mengen und Relationen
- Begriffe und Sätze der elementaren Zahlentheorie
- grundlegende algebraische Strukturen und ihre Eigenschaften

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Teilbarkeits-, Modulo- und Kongruenzberechnungen mit ganzen Zahlen und algebraischen Strukturen durchführen

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- logische Schlussfolgerungen nachzuvollziehen
- konkrete Anwendungen durch abstrakte mathematische Methoden zu analysieren und zu lösen

Inhalt:

- Beweistechniken, vollständige Induktion, Aussagenlogik,
- Mengenlehre, Relationen,
- Zahlentheorie: Teilbarkeit, Module, Kongruenz, Arithmetik, Division mit Rest, multiplikative Inverse, Primzahlen, Euklidischer Algorithmus, Kleiner Satz von Fermat, Eulersche Funktion, Diophantische Gleichungen, Großer Satz von Fermat, Chinesischer Restsatz,
- Algebraische Strukturen und Unterstrukturen: Monoide, Gruppen, Ringe, Körper, Ordnung, von Elementen, zyklische Gruppen, Generatoren, Vektorräume,
- Polynomringe und Galois Körper, Faltung
- Anwendungsbeispiele aus dem Bereich der symmetrischen und asymmetrischen Verschlüsselung sowie Protokollen der Rechnerkommunikation werden exemplarisch behandelt.

Literaturhinweise:

- J. Koch, M. Stämpfle: Mathematik für das Ingenieurstudium, 4. Auflage, Hanser Verlag 2017, ISBN 9783446451667

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur 90 Minuten
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Mathematik 2

Schlüsselworte: Differentialgleichungen, Differenzgleichungen, Potenzreihen, Fourier-Reihen

Zielgruppe:	2. Semester SWB	Modulnummer:	SWB 105 2003
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		75 h
	Selbststudium		45 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Jürgen Koch		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Lineare Gleichungssysteme, Vektoren, Matrizen, Funktionen in einer und in mehreren reellen Veränderlichen, komplexe Zahlen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, naturwissenschaftliche und technische Problemstellungen mathematisch zu lösen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die wichtigsten Begriffe und Eigenschaften von Differentialgleichungen, Differenzgleichungen, Potenzreihen und Fourier-Reihen

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Differentialgleichungen und Differenzgleichungen lösen
- Funktionen als Potenzreihen darstellen
- Periodische Funktionen durch Fourier-Reihen analysieren

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- Problemstellungen systematisch zu analysieren und zu lösen
- logische Schlussfolgerungen nachzuvollziehen

Inhalt:

- Lineare und nichtlineare Differentialgleichungen
- Lineare Differentialgleichungssysteme
- Lineare Differenzgleichungen und Differenzgleichungssysteme
- Potenzreihen und Taylor-Reihen
- Fourier-Reihen

Literaturhinweise:

- J. Koch, M. Stämpfle: Mathematik für das Ingenieurstudium, 4. Auflage, Hanser Verlag 2017, ISBN 9783446451667

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur 90 Minuten
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können Problemstellungen aus Naturwissenschaft und Technik mithilfe mathematischer Modelle am Computer lösen, simulieren und visualisieren.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Mensch-Computer-Interaktion 1

Schlüsselwörter: Theorie und Praxis von User Interfaces

Zielgruppe:	2. Semester SWB	Modulnummer:	IT 105 2001
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Astrid Beck		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

keine

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben die Kompetenz, Konzepte der Mensch-Computer-Interaktion anzuwenden. Sie werden zur Konzeption und Gestaltung benutzerfreundlicher interaktiver Applikationen befähigt. Sie erwerben ein grundlegendes Verständnis über benutzerfreundliche Interaktionen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Softwareergonomie,
- wahrnehmungspsychologische Grundlagen,
- Typographie

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- Anforderungsanalysen vorzunehmen,
- Dialogelemente sinnvoll anwenden,
- Prototypen für Usability-tests erstellen,
- Informationsarchitekturen benutzerfreundlich umsetzen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- gebrauchstaugliche Software zu konzipieren und umzusetzen, die effizient, effektiv vom Benutzer eingesetzt werden kann und zur Zufriedenheit führt.

Inhalt:

- Vorgehensmodell für die benutzerorientierte Systementwicklung
- Anforderungsermittlung, Prototyping, Usability Test, Benutzerprofile
- Softwareergonomische und wahrnehmungspsychologische Grundlagen
- Benutzergerechte Gestaltung von Dialogen, Anwendung von Dialogelementen
- Grundkenntnisse zu Typografie und Farbgestaltung
- Informationsarchitektur, Visualisierung und Navigation
- Aktuelle Fragestellungen, z.B.: Interkulturelle Gestaltung, Accessibility, Gestaltung mobiler Systeme, Gestaltung im Automotive Bereich

Literaturhinweise:

- Dahm: Grundlagen der Mensch-Computer-Interaktion, Pearson, 2006.
- Heinecke: Mensch-Computer-Interaktion, Springer, 2004.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden sind in der Lage, benutzerfreundliche Interaktionen im Rahmen eines Projektes zu bewerten und zu erstellen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Objektorientierte Systeme 1

Schlüsselwörter: Objektorientierte Programmierkonzepte

Zielgruppe:	2. Semester SWB	Modulnummer:	IT 105 2027
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		75 h
	Selbststudium		45 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Andreas Rößler		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse einer Programmiersprache

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben eine fundierte Grundlagenausbildung in Informatik und Programmieren. Sie beherrschen die Programmiersprache C++.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Klassenkonzepte
- Operatoren und Overloading
- Vererbung und Polymorphie

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- abstrakte Klassen erstellen
- Programme in C++ erstellen

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- Programme in C++ methodisch zu programmieren

Inhalt:

Es werden grundlegende Konzepte der objektorientierten Programmierung vermittelt. Hierzu gehören:

- Klassenkonzept (Attribute, Methoden), Information-Hiding (public, private),
- Konstruktoren und Destruktoren
- Statische Variablen und statische Methoden
- Operatoren und Overloading
- Vererbung und Polymorphie
- Abstrakte Klassen und ihre Rolle als Schnittstellendefinition

Als weitere Themen, die bei der objektorientierten Software-Entwicklung wichtig sind, werden behandelt:

- Referenzen, Namensräume, Umgang mit Strings
- Definition und Behandlung von Ausnahmen
- Bearbeitung von Dateien mit Hilfe von Streams
- Cast-Operatoren und die Typbestimmung zur Laufzeit

Literaturhinweise:

- Bjarne Stroustrup: Einführung in C++, Pearson Verlag, 2010, ISBN 9783868940053.
- Jürgen Wolf: C++, Galileo Computing, 2014, ISBN 978-3-8362-3895-3.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden beherrschen die methodische Programmierung objektorientierter Systeme.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Statistik

Schlüsselwörter: Kombinatorik, Wahrscheinlichkeitsrechnung, Statistik

Zielgruppe:	2. Semester SWB	Modulnummer:	IT 105 2018
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		75 h
	Selbststudium		45 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Gabriele Gühring		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Funktionen in einer und in mehreren reellen Veränderlichen, Matrizenrechnung

Gesamtziel:

Die Studierenden werden in die Lage versetzt, zufällige und mit Unsicherheit behaftete Phänomene zu beschreiben, zu erklären und zu verstehen. Sie beherrschen die grundlegenden Methoden der Wahrscheinlichkeitsrechnung, Statistik und Kombinatorik.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die grundlegenden kombinatorischen Formeln und ihre Anwendbarkeit auf entsprechende Fragestellungen,
- die grundlegenden wahrscheinlichkeitstheoretischen Kennzahlen und ihre Berechnungen bzw. Beziehungen untereinander,
- die grundlegenden statistischen diskreten und stetigen Verteilungen
- die Grundlagen der beschreibenden Statistik und der schließenden Statistik und können sie auf spezifische Situationen anwenden.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- große Datensätze zu beschreiben und Informationen darzustellen
- Ereignisse mit Häufigkeiten, Mittelwert und Varianz bzw. Standardabweichung zu beschreiben
- Aussagen über mit Unsicherheit behaftete Probleme zu bewerten und einzuordnen

Übergreifende Kompetenzen

Die Studierenden können

- Aussagen über mit Unsicherheit behaftete Fragestellungen herleiten, bewerten, einordnen
- Statistik als wichtiges Instrument zur Unterstützung der Arbeit mit großen Datenmengen

Inhalt:

- Datengewinnung und Datenbereinigung
- Darstellung statistischen Materials (Merkmaltypen, grafische Darstellung, Lageparameter einer Stichprobe)
- Mehrdimensionale Stichproben (Korrelation und Regression)
- Kombinatorik
- Wahrscheinlichkeitsrechnung (Laplace-Modelle; Zufallsvariablen und Verteilungsfunktionen; spezielle Verteilungsfunktionen wie z. B. Normal- oder Binomialverteilung)
- Schließende Statistik, insbesondere statistische Testverfahren und Vertrauensbereiche, p-Wert
- Einführung in stochastische Prozesse

Literaturhinweise:

- L. Sachs: Angewandte Statistik: Methodensammlung mit R, Springer Verlag, 16. Auflage 2018, ISBN 3662566567
- S. Ross: Statistik für Ingenieure und Naturwissenschaftler, 3. Auflage, Spektrum Verlag, 2006, ISBN 3827416213

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur 90 Minuten
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden beherrschen eine Anwendungssoftware, mit der sie statistische Fragestellungen auswerten und darstellen können.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Datenbanken 1

Schlüsselwörter: SQL, ODBC, Transaktionen, DBMS-Administration

Zielgruppe:	3. Semester SWB	Modulnummer:	IT 105 3007
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		75 h
	Selbststudium		45 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch und Englisch		
Modulverantwortung:	Prof. Jürgen Nonnast		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse in Betriebssystemen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erlernen die Grundkonzepte von hierarchischen netzwerkorientierten, relationalen und objektorientierten Datenmodellen. Sie sind in der Lage, Datenbank-anwendungen zu entwickeln. Sie können Datenbank-Anwendungen nach Vorgaben entwickeln. Sie beherrschen die Konzepte der Funktionsweise und des Betriebs von Datenbank-Managementsystemen und können diese bewerten.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Algebraische Relationen
- SQL-Funktionen

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- Embedded SQL mit C anwenden,
- Verknüpfungen von Tabellen erstellen
- DML- und DDL-Zugriffe durchführen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- ein Datenbank-Managementsystem zu konfigurieren.

Inhalt:

- Grundlagen von Datenmodellen
- Relationen Algebra
- SQL: Projektion, Restriktion, Unterabfragen, Skalare Funktionen, Aggregatfunktionen
- Datumsfunktionen
- DML-Zugriffe und DDL-Zugriffe
- Verknüpfung von Tabellen (Inner, Left, Right, Outer Join)
- Embedded SQL mit C (Singleton Select, Cursor Select, Cursor Update)
- Betrachtungen zur portablen Applikationsentwicklung mit SQL99
- Aufbau und Funktionsweise eines Datenbank-Managementsystems mit besonderem Fokus auf Mehrbenutzerbetrieb und Performance, Datensicherheit, Verfügbarkeit

Literaturhinweise:

- Baklarz, Zikopoulos: DB2 9 DBA Guide, Reference, and Exam Prep, IBM Press, 2007.
- E. Sanders: DB2 9 Fundamentals: Certification Study Guide, MC Press Online, 2007.
- E. Sanders: DB2 9 Database Administration: Certification Study Guide MC Press Online, 2007.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden können Betriebskonzepte nach Vorgabe realisieren.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Internet Technology

Keywords: Internet, Web, Client, Server, HTTP, HTML, CSS, Javascript, PHP

Audience: 4. Semester SWB **Module Number:** IT 105 3010

Workload: 5 ECTS **150 h**
divided into
Contact time 90 h
Self-study 30 h
Exam preparation 30 h

Course language: English

Modul director: Prof. Dr. -Ing. Harald Melcher

Valid from: 01.09.2019

Recommended requirements:

Knowledge in an object oriented programming language like Java or C#. Routine in a development IDE like IntelliJ or VisualStudioCode.

Desired learning outcomes of the module:

Students are proficient in selecting the right tools for Web based client server applications. They know the security risks and how to mitigate them and they have a basic understanding of the programming languages in use for Web applications.

Knowledge - professional competences

- Students acquire knowledge in the area of web based applications and services. They gain an overview over the protocols, the interworking of clients and servers and the major languages of the internet.

Skills - methodical competences

- Students are able to appraise the best combination of technologies for a specific web task. They can estimate the risk of a given solution.

Comprehensive Competencies

- Students understand, how web based services interact and are able to develop a simple service by themselves.

Contents:

- Basic structure of client – server communication
- Basic functions of a web server
- The web protocol HTTP
- Use of markup languages like HTML or XML
- Design and implementation of interactive web applications with HTML, CSS, Javascript and JSON

Literature:

- Freeman & Robson, Head First HTML5 Programming, O'Reilly
- Freeman & Robson, Head First HTML and CSS, O'Reilly
- Crockford, Javascript: The good Parts, O'Reilly
- Chaffer & Swedberg, Learning jQuery, Packt Publishing
- Bibeault & Katz, jQuery in Action, Manning

Offered:

Each semester

Sumodules and Assessment:

Type of instruction: Lecture with exercises and exam preparation
Type of assessment: Exam (90 minutes)
Hours per week: 3 SWS
Estimated student workload: 120 Hours

Type of instruction: Lab Work
Type of assessment: Report and Presentation
Hours per week: 1 SWS
Estimated student workload: 30 Hours

Learning outcomes:

Students are proficient in developing simple Web Applications according to best practice examples. They have experienced the pitfalls of Javascript and CSS programming and know how to cope with them.

Generation of the module grade:

Exam graded
Report and Presentation ungraded

Modulbeschreibung Modellbildung und Simulation

Schlüsselwörter: Mathematische Modelle, Simulation, Numerische Verfahren

Zielgruppe:	3. Semester SWB	Modulnummer:	SWB 105 3003
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Jürgen Koch		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse der Stochastik und der Mathematik, insbesondere Aufstellen von Differentialgleichungen

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden befähigt, eine technische Problemstellung in ein mathematisches Modell zu übertragen. Sie können dieses Modell in eine Simulation überführen sowie die Simulationsergebnisse bewerten und auf deren Grundlage die Modellbildung optimieren. Sie kennen die theoretischen Grundlagen der Modellbildung und können Qualität und Grenzen eines mathematischen Modells beschreiben und bewerten. Sie können die Dynamik einfacher physikalischer oder technischer Systeme mathematisch beschreiben und eine numerische Simulation dieses Modells konzipieren und implementieren. Sie sind in der Lage, die Aussagekraft von Simulationen zu bewerten und Simulationen im Hinblick auf die Qualität der Ergebnisse einerseits und den damit verbundenen Aufwand andererseits zu optimieren.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Die Methoden der mathematischen Modellierung,
- numerische Lösungen mathematischer Modelle.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- diskrete und kontinuierliche Systeme modellieren,
- Verknüpfungen von Tabellen erstellen,
- DML- und DDL-Zugriffe durchführen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- technische Problemstellung als ein mathematisches Modell darstellen und dieses in eine Simulation überführen,
- die Qualität der Simulation zu bewerten.

Inhalt:

- Modellbildung, Tragweite und Grenzen mathematischer Modelle
- Methoden der mathematischen Modellierung
- Numerische Lösung mathematischer Modelle
- Modellierung und Simulation diskreter Systeme (z.B. Entscheidungsmodelle, Reihenfolgeprobleme)
- Modellierung und Simulation kontinuierlicher Systeme (z.B. Populationsdynamik, Fluidströmungen)
- Aufwand und Präzision numerischer Simulationen
- Determinismus und chaotisches Verhalten

Literaturhinweise:

- Bungartz, Zimmer, Buchholz, Pflüger: Modellbildung und Simulation - eine anwendungsorientierte Einführung, Springer, 2009.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden können einfache Methoden zur numerischen Lösung von linearen Gleichungssystemen und Differentialgleichungen implementieren. Sie sind in der Lage, mathematische Modelle mittels angemessener rechnergestützter Verfahren numerisch zu lösen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Objektorientierte Systeme 2

Schlüsselwörter: Programmierparadigmen, Bibliotheken, Grafische Oberflächen

Zielgruppe:	3. Semester SWB	Modulnummer:	SWB 105 3029
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Kai Warendorf		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse einer objektorientierten Programmiersprache

Modulziel – angestrebte Lernergebnisse:

Die Studierenden vertiefen die objektorientierten Programmierparadigmen und deren praktische Anwendung. Sie können unterschiedliche Programmierparadigmen anwenden, Bibliotheken erstellen und verwenden sowie grafische Oberflächen aufbauen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- generische, parallele und funktionale Programmierung,
- Bibliotheken,
- Grafische Oberflächen.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- grafische Oberflächen erstellen,
- das Layoutmanagement durchführen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- Programmierparadigmen sowie graphische Oberflächen anwenden.

Inhalt:

Programmierparadigmen:

- Parallele Programmierung
- Funktionale Programmierung
- Generische Programmierung
- Bibliotheken
- Grafische Oberflächen
- Layout Management
- Eventhandling

Literaturhinweise:

- Paul Deitel, Java How to Program: Late Objects Version, Prentice Hall, 2010.
- Bernd Oestereich: Objektorientierte Softwareentwicklung. Analyse und Design mit UML 2.1, Oldenbourg Verlag, 2006.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lernziele:

Die Studierenden vertiefen und festigen ihre Fähigkeiten von Programmierparadigmen sowie vom Aufbau graphischer Oberflächen.

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden sind in der Lage, Konzepte der parallelen und graphischen Programmierung unter Anwendung professioneller Produktionswerkzeuge selbstständig umzusetzen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Rechnernetze

Schlüsselworte: Netztechnik, Protokolle, Ethernet, TCP/IP

Zielgruppe: 3. Semester SWB **Modulnummer:** IT 105 3008

Arbeitsaufwand:	5 ECTS	150 h
Davon	Kontaktzeit	75 h
	Selbststudium	45 h
	Prüfungsvorbereitung	30 h
Unterrichtssprache:	Deutsch	
Modulverantwortung:	Prof. Dr.-Ing. Michael Scharf	

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Kompetenzen in den Bereichen Programmierung und Betriebssysteme

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben Kenntnisse über grundlegenden Konzepte und Technologien in Rechnernetzen. Sie können die grundlegenden Konzepte von Rechnernetzen beschreiben. Sie verstehen das Schichtmodell in Kommunikationsnetzen und die Grundmechanismen und Aufgaben von Kommunikationsprotokollen. Die Funktionsweise wichtiger Standards wie Ethernet und TCP/IP sind den Studierenden bekannt. Dies ermöglicht es ihnen, geeignete Lösungen für verschiedene Anwendungszwecke auszuwählen und zu bewerten.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- den Aufbau von Kommunikationsnetzen und das Schichtenmodell,
- die Grundmechanismen und Aufgaben von Protokollen,
- die prinzipielle Arbeitsweise wichtiger Standards wie Ethernet und TCP/IP,
- die Funktionen, Komponenten und Dienste moderner Rechnernetze.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Kommunikationsdienste zu konfigurieren,
- bestehende Netztechnik und Protokolle zu analysieren,
- Kommunikationsmechanismen gezielt und sinnvoll einzusetzen.

Übergreifende Kompetenzen

Die Studierenden können

- das Zusammenspiel von Rechnernetzen, Betriebssystemen und Anwendungen beschreiben.

Inhalt:

- Grundlagen und Netzarchitekturen
- Kommunikation in lokalen Netzen
- Paketvermittlung im Internet
- Transportprotokolle im Internet
- Internet-Anwendungen
- Technologien in lokalen Netzen
- Technologien in Weitverkehrsnetzen

Literaturhinweise:

- Tanenbaum, Wetherall: Computernetzwerke, Pearson, 2012
- Kurose, Ross: Computernetzwerke: Der Top-Down-Ansatz, Pearson, 2014

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübungen
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können Netzwerkdienste konfigurieren, Kommunikationsprotokolle nutzen und deren Funktion analysieren und gegebenenfalls Fehler finden.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Softwaretechnik

Schlüsselworte: Software Engineering, Modellierung, Qualitätssicherung

Zielgruppe:	3. Semester	Modulnummer:	IT 105 3039
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		75 h
	Selbststudium		45 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr. Mirko Sonntag		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse einer höheren Programmiersprache

Modulziel – angestrebte Lernergebnisse:

Die Studierenden verstehen und beherrschen ingenieurmäßiges Software-Engineering. Die Studierenden verfügen über Wissen in den Bereichen ingenieurmäßige Software-Entwicklung, Vorgehensmodelle, Anforderungsanalyse, Qualitätssicherung, Modellierung und Versionsverwaltung.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Die Notwendigkeit für ingenieurmäßige Software-Entwicklung
- Plangetriebene und agile Vorgehensmodelle zur Software-Entwicklung
 - Phasen, Meilensteine und Artefakte
 - Rollen und Aufgaben
- Methoden zum Aufnehmen von Anforderungen
- Software-Spezifikation und -Entwurf
- Maßnahmen zur Sicherung der Software-Qualität
- Versionsverwaltung und Konfigurationsmanagement

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- zwischen einem plangetriebenen oder agilen Vorgehensmodell zu entscheiden
- planvoll Anforderungen aufzunehmen und zu dokumentieren
- eine Software-Spezifikation und einen Software-Entwurf zu erstellen
- IT-Projekte durchzuführen, die eine hohe Software-Qualität sicherstellen
- mit einer Versionsverwaltung umzugehen

Übergreifende Kompetenzen

Die Studierenden können

- Methoden des Software Engineering anwenden und damit ein IT-Projekt durchführen

Inhalt:

- Prinzipien des Software Engineering
- Plangetriebene und agile Vorgehens- und Prozessmodelle
- Requirements Engineering
- Systemspezifikation
- Systementwurf
- UML
 - Modellelemente: Knoten, Kanten, Beschriftungen
 - Beziehungen: Assoziation, Multiplizität, Qualifizierung, Generalisierung, Aggregation und Komposition
 - Use Case-, Klassen-, Objekt-, Sequenz-, Aktivitäts- und Zustandsdiagramme
- Versionsverwaltung und Konfigurationsmanagement
- Software-Qualität, Einführung in Software-Testing
- Software-Projektmanagement

Literaturhinweise:

- Ludewig and Lichter: Software Engineering, 2007, dpunkt.
- Sommerville: Software Engineering, 2011, Addison-Wesley.
- Brügge and Dutoit: Object-Oriented Software Engineering, 3rd edition, 2010, Prentice Hall.
- Baumgartner et al.: Agile Testing, 2018, Hanser.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden beherrschen die Methoden agile Software-Entwicklung, Requirements Engineering, Modellierung mit UML, Unit-Testing und Versionsverwaltung.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Blockseminar Software-Projekt Management
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden erlernen das erfolgreiche Durchführen von Projekten. Sie beherrschen die Instrumente des Projektmanagements.

Bildung der Modulnote:

Klausur, 2 unbenotete Testate

Modulbeschreibung Algorithmen und Datenstrukturen

Schlüsselworte: Algorithmen, Datenstrukturen, Graphen

Zielgruppe:	4. Semester SWB	Modulnummer:	IT 105 3012
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Jürgen Koch		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

- Mathematik 1 und 2,
- Programmieren, Objektorientierte Systeme

Modulziel – angestrebte Lernergebnisse:

Die Studierenden besitzen einen Überblick über die wichtigsten Klassen von Algorithmen. Sie können grundlegende Merkmale, Leistungsfähigkeit, Gemeinsamkeiten und Querbezüge unterschiedlicher Algorithmen beurteilen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die wesentlichen Algorithmen.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können

- die Komplexität von Algorithmen einschätzen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- grundlegende Algorithmen und Datenstrukturen bezüglich ihrer Eigenschaften und Leistungsfähigkeit richtig anwenden und einschätzen.

Inhalt:

- Darstellung, Design und Klassifikation von Algorithmen
- Einfache und abstrakte Datenstrukturen: Arrays, Listen, Mengen, Verzeichnisse
- Komplexität, Effizienz, Berechenbarkeit, O-Notation
- Such- und Sortierverfahren
- Bäume und Graphen
- Iterative Verfahren (Gauß, Newton)
- Hash-Verfahren
- Geometrische Algorithmen
- String-Matching Algorithmen und endliche Automaten
- Zufallszahlen und Monte Carlo Algorithmen

Literaturhinweise:

- Robert Sedgewick, Algorithmen in C++, Addison-Wesley
- G. Saake, K. Sattler: Algorithmen und Datenstrukturen, dpunkt.verlag
- G. Pomberger, H. Dobler: Algorithmen und Datenstrukturen, Pearson Studium

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Computerarchitektur

Schlüsselworte: Rechnerarchitektur, Mikroprozessor, Mikrocontroller, Instruction Set Architecture, Assemblerprogrammierung

Zielgruppe: 4. Semester SWB **Modulnummer:** IT 105 4003

Arbeitsaufwand: 5 ECTS **150 h**
Davon
Kontaktzeit **75 h**
Selbststudium **45 h**
Prüfungsvorbereitung **30 h**

Unterrichtssprache: Deutsch oder Englisch
Modulverantwortung: Prof. Dr.-Ing. Werner Zimmermann

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Programmieren, Digitaltechnik 1 – 2, Softwaretechnik, Informationstechnik, Betriebssysteme

Modulziel – angestrebte Lernergebnisse:

Die Studierenden sind in der Lage, den Aufbau und die Funktionsweise von Mikroprozessoren sowie ihre Peripheriebausteine zu verstehen und zu programmieren. Sie beherrschen ein Grundverständnis für die Instruction Set Architecture von Rechnern und verstehen, wie die Programmierkonstrukte höherer Programmiersprachen auf die "Sprache der Hardware" abzubilden sind. Sie verstehen das Zusammenwirken von Programmiersprache, Betriebssystem und Hardware, um effizientere Software zu entwickeln.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Architektur von Rechnersystemen mit Mikroprozessoren und Mikrocontrollern,
- die Maschinenbefehlsebene (Instruction Set Architecture) von Rechnern.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Programmierkonstrukte höherer Programmiersprachen auf die "Sprache der Hardware" abgebildet werden,
- Programme in Assembler erstellen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- das Zusammenwirken von Programmiersprache, Betriebssystem und Hardware abzubilden.

Inhalt:

- Aufbau von Rechnersystemen, arithmetisch-logische Operationen, Grundaufgaben von Betriebssystemen
- Programmiermodell (Registersatz, Adressierungsarten, Memory Map, Befehlssatz) eines beispielhaften Mikroprozessors
- Einführung in die Maschinensprache, Abbildung wichtiger Hochsprachenkonstrukte auf die Maschinensprache, Abschätzung des Speicherplatzbedarfs und der Ausführungsgeschwindigkeit
- Hardware/Softwareschnittstelle für typische Peripheriebausteine, digitale und analoge Ein-/Ausgabe, Timer, einfache Netzwerkschnittstellen
- Modulare Programmierung, Schnittstellen für das Zusammenspiel verschiedener Programmiersprachen
- Unterstützung von Betriebssystem-Mechanismen, z.B. Speicherschutz, virtueller Speicher, durch Mikroprozessoren
- Überblick über aktuelle Mikro- und Signalprozessorarchitekturen: Technik und Marktbedeutung

Literaturhinweise:

- Patterson, D.; Hennesey, J.: Computer Architecture and Design. Morgan Kaufmann Verlag, 2008.
- Tanenbaum, A.: Structured Computer Organization. Prentice Hall Verlag, 2012.
- Huang, H.W.: The HCS12/9S12. An Introduction to the hardware and software interface. Thomson Learning Verlag, 2009.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können die Grundlagen der hardwarenahen Programmierung in C/C++ und Maschinensprache (Assembler) in praktischen Übungen umsetzen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Datenbanken 2

Schlüsselworte: Datenbanken, DBMS

Zielgruppe: 4. Semester SWB
4. Semester WKB **Modulnummer:** IT 105 4018

Arbeitsaufwand: 5 ECTS **150 h**
Davon **Kontaktzeit** 60 h
Selbststudium 60 h
Prüfungsvorbereitung 30 h

Unterrichtssprache: Deutsch
Modulverantwortung: Prof. Dr. Dirk Hesse

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Datenbanken 1

Modulziel – angestrebte Lernergebnisse:

Mit diesem Modul vertiefen die Studierenden ihre Kenntnisse auf dem Gebiet der Datenbanktechnologien. Vermittelt wird die Fähigkeit, Datenbanken zu entwickeln und die referentielle Integrität sicherzustellen. Die Studierenden verstehen die physische Organisation der Daten und kennen die darauf aufbauende logische und physische Optimierung. Sie können anwendungsbezogen passende Speicherarrays unterscheiden und auswählen. Sie beherrschen die verschiedenen Indexstrukturen und können in Anhängigkeit vom Abfrageverhalten geeignete Indizes auswählen und einrichten. Sie sind in der Lage kostenbasierte Optimierungen durch Anwendung äquivalenzerhaltender Transformationsregeln durchzuführen.

Vermittelt werden weiterhin Kenntnisse über neue Entwicklungen im Bereich der Hauptspeicherdatenbanken.

Die Studierenden sind in der Lage eine Datenbankverbindung mit einem Anwendungsprogramm zu programmieren und kennen die Grundlagen der Datenbankanbindung über das Internet.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Entwicklungszyklus einer DB
- UML, ERM Notationsformen, Normalformen
- Physische Datenorganisation einer DB, Speicherorganisation, Indexstrukturen
- Äquivalenzerhaltende Transformationsregeln
- Physische Umsetzung der logischen Algebra
- Kostenmodelle und die Grundlagen kostenbasierter Optimierung
- Neue Konzepte von In-Memory-Datenbanken

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- ERM Modelle zu entwickeln
- Referentielle Integrität und komplexe Integrationsbedingungen sicherzustellen
- Geeignete Indexstrukturen auszuwählen und einzurichten
- Äquivalenzerhaltende Transformationsregeln zur kostenbasierten Optimierung anzuwenden
- Kostenmodelle (QEP) zu lesen und dementsprechende Abfrageoptimierungen durchzuführen

Übergreifende Kompetenzen

Die Studierenden können

- Anspruchsvolle SQL Aufgaben selbständig lösen und Programmierung in SQL durchzuführen
- Geeignete Methoden der Abfrageoptimierung im Systemzusammenhang anwenden
- Erlangte Kompetenzen im Rahmen von DB-Programmierung und Java-Anwendungsprogrammierung umsetzen

Inhalt:

- Advanced SQL und SQL Programmierung
- DB Design: Development Lifecycle, ERM, Normalization
- Datenintegrität, referentielle Integrität in SQL, komplexe Integritätsbedingungen, Trigger
- Physische Datenorganisation: Speicherarrays, Page-Frames, Indexstrukturen, ISAM, B-Bäume, dynamisches und statisches Hashing, mehrdimensionale Indexstrukturen
- Anfrageoptimierung: Logische Optimierung, Äquivalenzen in der relationalen Algebra, Anwendung äquivalenzerhaltender Transformationsregeln
- Physische Optimierung: Implementierung von Selektion, Projektion und Vereinigung, Sort-Algorithmen, Übersetzung der logischen Algebra
- Kostenmodelle und Selektivität, „Tuning“, kostenbasierte Optimierung
- Hauptspeicherdatenbanken: Entwicklungen, Einsatzbereiche, Datenstrukturen
- DB-Programmierung: Stored Procedure, Function, Trigger, Cursor
- eingebettetes SQL, JDBC, ODBC, Verbindungsaufbau und Programmbeispiele

Literaturhinweise:

- Kemper, Alfons: Datenbanksysteme. 10. Auflage 2015.
- Conolly, Thomas: Database Systems. 6. Auflage 2014.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	KL 90 Minuten
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können

- anspruchsvolle SQL Aufgaben selbständig lösen und Programmierung in SQL durchführen,
- komplexe Integritätsbedingungen durch SQL Methoden sicherstellen,
- geeignete Methoden der Abfrageoptimierung und der kostenbasierten Optimierung im Systemzusammenhang anwenden,
- erlangte Kompetenzen im Rahmen von SQL Programmierung und Java-Anwendungsprogrammierung umsetzen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Projekt Softwaretechnik

Schlüsselwörter: Teamprojekt, Projektmanagement, Softwareentwicklung

Zielgruppe:	4. Semester SWB	Modulnummer:	IT 105 4026
Arbeitsaufwand:	10 ECTS		300 h
Davon	Kontaktzeit		60 h
	Selbststudium		240 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Andreas Rößler		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse über Programmiersprachen und Methoden der Softwaretechnik

Gesamtziel:

Die Studierenden erwerben die Kompetenz zur Entwicklung komplexer Software-Anwendungen.

Die Studierenden können das bereits erworbene Wissen im Kontext einer interaktiven, multimedialen Software-Entwicklungsaufgabe anwenden und vertiefen. Sie beherrschen die methodische Vorgehensweise der Software-Entwicklung. Des Weiteren sind sie in der Lage, Methoden und Techniken aus dem Bereich Soft Skills anzuwenden.

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben die Kompetenz zur Entwicklung komplexer Software-Anwendungen. Sie können das bereits erworbene Wissen im Kontext einer interaktiven, multimedialen Software-Entwicklungsaufgabe anwenden und vertiefen. Sie beherrschen die methodische Vorgehensweise der Software-Entwicklung. Des Weiteren sind sie in der Lage, Methoden und Techniken aus dem Bereich Soft Skills anzuwenden.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Methoden der Software-Entwicklung,

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- die Methoden des Projektmanagement anwenden, die im Studium erlernten Modelle und Methoden in der Software-Entwicklung einsetzen.
- das erarbeitete Ergebnis vor einer Gruppe sicher präsentieren.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- methodische Vorgehensweisen der professionellen Software-Entwicklung einsetzen.

Inhalt:

- Projektmanagement und Teamarbeit
- Arbeitstechniken:
Zeitmanagement, Arbeitsorganisation, Informationsgewinnung/-recherche
- Wissenschaftliches Arbeiten
- Kommunikation und Präsentation
- Technische Dokumentation
- Softwaretechnik:
Anforderungsanalyse, Design, Implementierung, Test, Installation

Literaturhinweise:

- Ludwig, Richter: Software Engineering. dpunkt Verlag, 2013.
- Kraus, Georg Westermann, Reinhold: Projektmanagement mit System, Vieweg Verlag, 2010.

Wird angeboten:

in jedem Semester

Lehr- und Lernform:	Teamprojekt
Leistungskontrolle:	Bericht und Referat (20 Minuten)
Anteil Semesterwochenstunden:	8 SWS
Geschätzte studentische Arbeitszeit:	240 Stunden

Lernziele:

Die Studierenden können methodische Vorgehensweisen der professionellen Software-Entwicklung einsetzen.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Lernziele:

Die Studierenden beherrschen Vorgehensweisen zur Verbesserung der persönlichen Fertigkeiten. Die Studierenden verfügen über Kenntnisse zur Rollenverteilung im Projekt-Team und der Gruppendynamik im Projekt-Team. Sie können das erarbeitete Ergebnis vor einer Gruppe sicher präsentieren.

Bildung der Modulnote:

Bericht und Referat benotet, unbenotetes Testat

Modulbeschreibung Softwarearchitektur

Schlüsselworte: Architekturen, Objektorientierte Modellierung

Zielgruppe: 4. Semester SWB **Modulnummer:** IT 105 4007

Arbeitsaufwand:	5 ECTS	150 h
Davon	Kontaktzeit	75 h
	Selbststudium	45 h
	Prüfungsvorbereitung	30 h
Unterrichtssprache:	Deutsch oder Englisch	
Modulverantwortung:	Prof. Dr. Jörg Friedrich	

Stand: 01.09.2019

Empfohlene Voraussetzungen:

- Objektorientierte Systeme
- UML 2

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, Anforderungen in komplexe Softwarearchitekturen umzusetzen. Sie können abgeleitete Anforderungen in Softwarearchitekturen umsetzen. Sie sind in der Lage, die passenden Entwurfs- und Architekturmuster sowie Frameworks und Bibliotheken einzusetzen. Sie besitzen die Kompetenz für ein ingenieurmäßiges Vorgehen bei der Erstellung der Software-Applikation.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Frameworks und Bibliotheken für SOA
- Entwurfsmuster

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Entwurfsmuster auswählen und anwenden,
- Webservices programmieren.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- Probleme im Bereich Softwarearchitektur zu lösen sowie die Auswahl von Software-Technologien zu bewerten.

Inhalt:

- Architektur und Architekten
- Vorgehen bei der Architekturentwicklung
- Architektursichten, UML 2 für Architekten
- Objektorientierte Entwurfsprinzipien
- Architektur- und Entwurfsmuster
- Technische Aspekte, Berücksichtigung von Anforderungen und Randbedingungen
- Middleware, Frameworks, Referenzarchitekturen, Modell-getriebene Architektur
- Komponenten, Komponententechnologien, Schnittstellen (API)
- Bewertung von Architekturen
- Refactoring, Reverse Engineering

Literaturhinweise:

- J. Goll: Methoden der Softwaretechnik, Vieweg-Teubner, 2012.
- J. Goll, M. Dausmann: Architektur- und Entwurfsmuster, Vieweg-Teubner, tbp 2013.
- G. Starke: Effektive Softwarearchitekturen, Hanser, 2011.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur 90 Minuten
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden können Entwurfs- und Architekturmuster auswählen und anwenden.
Sie sind in der Lage, Komponenten (EJB) sowie Webservices (SOA) zu programmieren.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Praktisches Studiensemester

Schlüsselwörter: Praktische Ingenieurserfahrung im industriellen Umfeld,
Projektarbeit im Team

Zielgruppe:	5. Semester WKB	Modulnummer:	IT 105 5000
Arbeitsaufwand:	26 ECTS		780 h
Davon	Kontaktzeit		780 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Kai Warendorf		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Abgeschlossener erster Studienabschnitt

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden zum ingenieurmäßigen Arbeiten auf dem Gebiet der Softwaretechnik befähigt. Die Studierenden beherrschen das ingenieurmäßige Arbeiten in einem Projektteam.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- den organisatorischen Aufbau und Funktionsweise einer Abteilung

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- die Methoden des Projektmanagement anwenden,
- die im Studium erlernten Modelle und Methoden zur Lösung berufspraktischer Problemstellungen anwenden die erlernten

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- sich im industriellen Umfeld einer Firma sicher zu bewegen,
- Lösungspraktiken der Praxis auf Basis der im Studium entwickelten Kompetenzen kritisch reflektieren.

Inhalt:

100 Tage betriebliche Praxis in einem Betrieb oder einer Firma aus dem IT-Bereich

Literaturhinweise:

- Lutz Hering, Heike Hering, Klaus-Geert Heyne: Technische Berichte, Vieweg, 2014.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Praktikum
Leistungskontrolle:	Bericht, Referat (20 Minuten)
Anteil Semesterwochenstunden:	26 SWS
Geschätzte studentische Arbeitszeit:	780 Stunden
Lernziele:	

Die Studierenden beherrschen das ingenieurmäßige Arbeiten in einem Projektteam.

Bildung der Modulnote:

unbenotetes Testat

Modulbeschreibung Schlüsselqualifikationen

Schlüsselworte: **Berufsstart, Wissenschaftliches Arbeiten, Disputation
Technisches Englisch**

Zielgruppe: **5. Semester WKB** **Modulnummer:** **IT 105 5001**

Arbeitsaufwand: **4 ECTS** **120 h**
Davon **Kontaktzeit** **60 h**
Selbststudium **60 h**

Unterrichtssprache: **Deutsch**
Modulverantwortung: **Prof. Dr.-Ing. Andreas Rößler**

Stand: **01.09.2019**

Empfohlene Voraussetzungen:

Schulkenntnisse in Englisch

Modulziel – angestrebte Lernergebnisse:

Studierenden erwerben Kompetenzen in

- Kommunikationsfähigkeit,
- Disputation,
- Fremdsprachen,
- wissenschaftlichen Schreiben,
- Bewerbungsverfahren.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- adäquates und situationsbezogenes berufliches Handeln.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- den Berufsstart erfolgreichen durchzuführen,
- sich sicher im beruflichen Umfeld bewegen und
- wissenschaftliche Artikel erstellen.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- wissenschaftliche Texte über ingenieurwissenschaftlich Themen auch in englischer Sprache zu erstellen,
- auch in englischer Sprache sicher zu kommunizieren.

Inhalt:

Wissenschaftliches Arbeiten

- Strukturieren
- Recherchieren
- Analysieren
- Wissenschaftliche Schreiben und Zitieren

Berufsstart

- Karriereplanung
- Bewerbertraining

Technisches Englisch

- TOEFL-Test

Literaturhinweise:

- B. Stemmer, T. Wynne: Grammar Rules. Grundlagen der englischen Grammatik, Klett Verlag, 2000.
- F. Schulz von Thun: Miteinander reden, Band 1-3, Rowohlt TB, 2008

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung und Übungen
Leistungskontrolle:	Hausarbeit und Referat (20 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lernziele:

Die Studierenden erwerben und vertiefen die Fähigkeit zur inhaltlichen Erfassung und Erstellung ingenieurwissenschaftlicher Texte.

Lehr- und Lernform:	TOEFL-Vorbereitungskurs
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden erwerben die Fähigkeit zur inhaltlichen Erfassung technisch-wissenschaftlicher Texte und zur Kommunikation über technisch- wissenschaftliche Themen in englischer Sprache.

Bildung der Modulnote:

unbenotetes Testat

Modulbeschreibung Informationssysteme

Schlüsselworte: Informationssysteme

Zielgruppe:	6. Semester WKB	Modulnummer:	IT 105 6001
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Dirk Hesse		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse in

- Programmieren
- Objektorientierte Systeme 1 und 2
- Softwaretechnik
- Datenbanken 1 und 2

Modulziel – angestrebte Lernergebnisse:

Vermittelt wird die Fähigkeit, Informationssysteme mit relationaler Datenhaltung von der Problemanalyse und Anforderungsdefinition über den Architekturentwurf bis zur Programmentwicklung und dessen Test zu entwerfen, zu entwickeln und zu betreiben. Die Studierenden verstehen die grundlegenden Konzepte des Entwurfs, der Entwicklung und des Betriebs von Informationssystemen. Sie können die Methoden und Vorgehensweisen der Informationssystemgestaltung mit den zugehörigen Schichtenarchitekturen und Datenmodellen in der Praxis anwenden. Sie sind in der Lage lauffähige Anwendungen mit Hilfe von Entwicklungsplattformen und CASE generierten Datenbankmodellen zu erstellen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Architekturen integrierter Informationssysteme
- UML, ERM, Datenmodelle, Normalformtheorien
- Architektur- und Datenbankmodelle im Rahmen des Architekturentwurfs
- Entwicklungswerkzeuge, Sprachen und Bibliotheken, Entwicklungsplattformen
- Datenbanksysteme und Datendienste

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Problemanalysen und Anforderungsdefinitionen durchzuführen
- Architektur- und Datenbankmodelle mit Hilfe von CASE Tools zu entwerfen
- Programmentwicklungen mit Hilfe von Entwicklungsplattformen durchzuführen
- Datenbanksysteme und Cloudanbindungen zu nutzen

Übergreifende Kompetenzen

Die Studierenden können

- Informationssysteme planen und entwickeln
- die Digitale Transformation durch Anwendung digitaler Technologien umsetzen

Inhalt:

- Problemanalyse und Anforderungsdefinition
- Architekturentwurf: Architekturmodelle und Schichtenmodelle
- Normalformtheorien
- Programmentwicklung und Test: Entwicklungswerkzeuge, Entwicklungsplattformen
- Nutzung von Datenbanksystemen und Datendiensten

Literaturhinweise:

- Wallace, P.: Introduction to Information Systems: People, Technology and Processes (3rd Edition), 2019
- Stair, Reynolds: Principles of Information Systems (Englisch) 13. Auflage, 2019
- Connolly, T.: Database Systems: A Practical Approach to Design, Implementation, and Management, 2014

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Workshop
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Lernergebnisse:

Die Studierenden beherrschen Analyse, Design und Implementierung einer Anwendung zur Ressourcenplanung.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung IT Security

Schlüsselwörter: Angriffe, Bedrohungen, Sicherheitsmaßnahmen, Kryptografie

Zielgruppe:	6. Semester SWB	Modulnummer:	SWB 105 6005
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		75 h
	Selbststudium		45 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch und Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Tobias Heer		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse in

- Rechnernetze,
- Programmieren,
- Lineare Algebra.

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben die Kompetenz zum sicheren Betrieb von Systemen der Informationstechnik. Sie sind in der Lage, die Risikobewertung und die Auswahl von Sicherheitsmaßnahmen in der Informationstechnik vorzunehmen. Sie besitzen die Fähigkeit, die Risikoeinschätzung vorzunehmen und abzuwägen. Des Weiteren verfügen sie über Kenntnisse zu sicheren Verschlüsselungsverfahren. Sie können Sicherheitsbeweise für Verschlüsselungsverfahren durchführen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Prinzipien von symmetrischer und asymmetrischer Verschlüsselung,
- können die Sicherheitsschwächen von IT-Systemen einschätzen
- Angriffe und Bedrohungen.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- die Sicherheit von Protokollen und Verschlüsselungsalgorithmen einzuschätzen
- sichere kryptografische Protokolle zu erstellen,
- Programme für sichere IT-Systeme zu erstellen,
- Sicherheitsmaßnahmen für IT-Systeme anzuwenden .

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- Sicherheitsschwächen bei IT-Systemen zu erkennen sichere IT-Systeme zu realisieren.

Inhalt:

- Grundbegriffe der IT-Sicherheit
- Sicherheitsschwächen in Netzwerkprotokollen
- Zugriffskontrolle auf Systeme
- Angriffe auf Systeme
- Programmieren für sichere Systeme
- Grundlagen der Kryptografie
- Moderne Verschlüsselungsverfahren
- Beweisbar sichere Verschlüsselung
- Kryptografische Sicherheitsdienste
- Authentifikationssysteme
- Methoden des Sicherheitsmanagements

Literaturhinweise:

- B. Schneier: Angewandte Kryptographie, Pearson Education Deutschland, 2005.
- B. Schneier: Applied Cryptography, John Wiley & Sons, Inc. 2015.
- M. Bishop: Computer Security: Art and Science, Addison Wesley Verlag, 2018.
- W. Stalling: Computer Security: Principles and Practice, Pearson Education, 2018.

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden sind in der Lage, Angriffsszenarien durchzuführen sowie Sicherheitsschwächen bei Netzwerkprotokollen zu erkennen. Sie können hierzu Abwehrmaßnahmen einsetzen und sichere Verschlüsselungsverfahren anwenden.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Parallele und Verteilte Systeme

Schlüsselwörter: Client/Server-Strukturen, Distributed Computing, Qualitätssicherung bei IT-Systemen

Zielgruppe:	6. Semester SWB	Modulnummer:	SWB 105 xxxx
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Rainer Keller		
Stand:	01.09.2019		

Empfohlene Voraussetzungen:

Kenntnisse in

- Programmieren,
- Rechnernetze,
- Softwarearchitektur.

Modulziel – angestrebte Lernergebnisse:

Studierende können die allgemeinen Anforderungen an Verteilte und parallele Systeme beschreiben. Sie sind in der Lage, verteilte Systeme mittels verschiedener, bestehender Technologien zu planen, erstellen, und zu evaluieren und zu Nutzen. Sie sind außerdem in der Lage, die Qualität von parallelen und verteilten Systemen zu beurteilen und geeignete Maßnahmen zur Qualitätssicherung solcher Systeme zu definieren und umzusetzen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Die Prinzipien paralleler und verteilter Systeme
- Die Technologien des verteilten und parallelen Rechnens
- Methoden zur Messung und Steigerung der Qualität bei IT-Systemen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- verteilte Systeme mit bestehenden Technologien erstellen
- Zugriffe auf lokale und entfernte Ressourcen zu vereinfachen
- Dienste spezifizieren

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- Verteilte und parallele Systeme sicher und redundant auszulegen und zu programmieren.

Inhalt:

- Motivation für Paralleles und Verteiltes Rechnen (Shared Memory, Message Passing, Shared Nothing)
- Grundlegende Technologien von verteilten Systemen und verteiltem Rechnen
- Komponenten Technologien
- Kommunikations-Methoden und Schnittstellen
- Service-orientierte Schnittstellen (REST) und MicroServices
- Evaluierung von Technologien
- Qualitätssicherung und Tools für Verteiltes Rechnen

Literaturhinweise:

- Schill, A., Springer, T.: Verteilte Systeme – Grundlagen und Basistechnologien, Springer Vieweg, 2. Auflage, 2012
- Tanenbaum, A., van Steen, M.: Verteilte Systeme: Prinzipien und Paradigmen, Pearson, 2. Auflage 2007
- Bengel, G. et al: Masterkurs Parallele und Verteilte Systeme, Springer Vieweg, 2. Auflage, 2015

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lehr- und Lernform:	Laborübung
Leistungskontrolle:	Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernziele:

Die Studierenden können Technologien für verteilte Systeme anwenden.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Software Testing

Schlüsselworte: Testen, Qualitätssicherung

Zielgruppe: 6. Semester SWB-SWT **Modulnummer:** IT 105 **NNNN**

Arbeitsaufwand: 5 ECTS **150 h**
Davon
Kontaktzeit **120 h**
Selbststudium **30 h**
Prüfungsvorbereitung **30 h**

Unterrichtssprache: Deutsch
Modulverantwortung: Prof. Dr. Mirko Sonntag

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Prinzipien des Software-Engineering und Kenntnisse in einer objektorientierten Programmiersprache.

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, durch qualitätssichernde Maßnahmen die Erfüllung der funktionalen und nicht-funktionalen Anforderungen an Software zu gewährleisten. Sie beherrschen Software-Tests als wichtigstes Mittel der Qualitätssicherung. Sie können Kosten, Nutzen und Grenzen von Software-Tests bei der Entwicklung von Test-Konzepten berücksichtigen und selbstständig Tests entwickeln.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Organisation und Management von Software-Tests
- Verschiedene Ebenen und Methoden von Software-Tests
- Nutzen und Kosten der Automatisierung von Tests
- Grenzen der Testautomatisierung

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Software-Tests auf verschiedenen Ebenen zu formulieren
- Unit Tests und Integrationstests zu entwickeln und Testobjekte zu isolieren
- Automatisierte Tests im Entwicklungsablauf einzubinden
- Systemtests toolgestützt durchzuführen

Übergreifende Kompetenzen

Die Studierenden können mithilfe von Tests sicherstellen, dass Software von hoher Qualität entwickelt wird.

Inhalt:

- Motivation für Qualitätssicherung und Testen
- Testautomatisierung, Testdokumentation, Testmanagement
- Testwerkzeuge
- Black Box und White Box Testing
- Unit Tests und zugehörige Methodiken wie Mocking und Test-driven Development
- Integrationstests
- System Tests (Performance & Load Testing, Penetration Tests)
- Akzeptanztests
- GUI Tests

Literaturhinweise:

- Baumgartner et al.: Agile Testing, Hanser, 2. Auflage, 2018
- Lisa Crispin, Janet Gregory: Agile Testing: A Practical Guide for Testers and Agile Teams, Addison-Wesley, 2008
- Eran Kinsbruner: The Digital Quality Handbook: Guide for Achieving Continuous Quality in a DevOps Reality, Infinity P, 2017
- Jonathan Rasmusson: The Way of the Web Tester, O'Reilly, 2016

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle: Klausur (90 Minuten)
Anteil Semesterwochenstunden: 2 SWS
Geschätzte studentische Arbeitszeit: 90 Stunden

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Laborübung
Leistungskontrolle: Testat
Anteil Semesterwochenstunden: 2 SWS
Geschätzte studentische Arbeitszeit: 60 Stunden

Lernziele:

Die Studierenden sind in der Lage automatische Tests auf verschiedenen Ebenen der Testpyramide zu implementieren und in die Continuous Integration Pipeline zu integrieren. Darüber hinaus können Sie toolgestützte Systemtests durchführen, um die Einhaltung nicht-funktionaler Anforderungen von Software sicherzustellen.

Bildung der Modulnote:

Klausur, unbenotetes Testat

Modulbeschreibung Sprachen und Automaten

Schlüsselworte: Formale Sprachen; Automatentheorie; P und NP

Zielgruppe: 6. Semester SWB-SWT **Modulnummer:** SWB 105 **xxxx**

Arbeitsaufwand: 5 ECTS 150 h
Davon
 Kontaktzeit 60 h
 Selbststudium 60 h
 Prüfungsvorbereitung 30 h

Unterrichtssprache: Deutsch
Modulverantwortung: Prof. Dr.-Ing. Steffen Schober

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Diskrete Mathematik

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erwerben das notwendige theoretische Wissen, um den Aufbau und Probleme formaler Sprachen und Automaten zu beschreiben. Sie verfügen über die Fähigkeit Techniken der Theoretischen Informatik in praktischen Anwendungen anzuwenden. Sie können damit verschiedene Fragestellungen und Probleme mit formalen Sprachen und den ihnen zugeordneten Automaten lösen und im Hinblick auf ihre Vollständigkeit, Abgeschlossenheit, Berechenbarkeit und Komplexität zu beurteilen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Grundbegriffe und mathematische Grundlagen
 - Formale Beweisverfahren
 - Aussage und Prädikatenlogik
- Grundbegriffe der Automatentheorie und der Formalen Sprachen:
 - Endliche Automaten und Reguläre Sprachen
 - Kellerautomaten und Kontextfreie Sprachen
 - Turingmaschinen und rekursiv aufzählbare Sprachen
- Grundbegriffe der Komplexitätstheorie:
 - P vs. NP sowie NP-Vollständigkeit

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage:

- den Aufbau formaler Sprachen zu beschreiben
- prinzipielle Schranken bestimmter Berechnungsmodelle zu verstehen
- Probleme mit formalen Sprachen und den ihnen zugeordneten Automaten zu lösen

Übergreifende Kompetenzen

Die Studierenden können

- Techniken der Theoretischen Informatik in praktischen Anwendungen nutzen, z.B. durch Nutzung von Parsergeneratoren
- Algorithmen und Spezifikationen formal beschreiben

Inhalt:

- Aussage- und Prädikatenlogik
- Formale Beweistechniken (insb. induktive und strukturelle Induktion)
- Chomsky-Sprachhierarchie und zugehörige Automatenmodelle
- Anwendung von Parser-Generatoren
- Komplexität von Entscheidungsproblemen
- Problemklassen P und NP sowie die NP-Vollständigkeit

Literaturhinweise:

- Hopcroft, u.a.: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson Studium
- Schöning: Theoretische Informatik – Kurzgefasst; Spektrum Akademischer Verlag

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Modulbeschreibung Studienprojekt

Schlüsselworte: Studienprojekt aus dem Gebiet der Softwaretechnik

Zielgruppe: 6. Semester SWB **Modulnummer:** IT 105 6007

Arbeitsaufwand: 5 ECTS **150 h**
Davon
Kontaktzeit **5 h**
Selbststudium **135 h**
Prüfungsvorbereitung **10 h**

Unterrichtssprache: Deutsch
Modulverantwortung: Prof. Dr.-Ing. Reinhard Schmidt

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Abgeschlossener erster Studienabschnitt

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt ein ingenieurwissenschaftliches Projekt auf dem Gebiet der Softwaretechnik zu bearbeiten.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die im Studium erlernten Modelle und Methoden zur Lösung ingenieurwissenschaftliche Problemstellungen anwenden.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Zeit- und Projektmanagement
- wissenschaftliches Arbeiten und Schreiben
- wissenschaftliches Präsentieren

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- selbstständig wissenschaftlich zu arbeiten.

Inhalt:

Im Studienprojekt ist unter Anleitung eines betreuenden Professors eine ingenieurmäßige Aufgabenstellung aus dem Gebiet der Softwaretechnik zu lösen.

Literaturhinweise:

- Lutz Hering, Heike Hering: Technische Berichte, Vieweg, 2017, ISBN 978-3-8348-15-86-6.
- Bernd Heesen; Wissenschaftliches Arbeiten, Springer Verlag, 2014, ISBN 978-3-662-43346-1,
- Henning Lobin; Die wissenschaftliche Präsentation: Konzept – Visualisierung – Durchführung; Schönigh Verlag, 2012, ISBN 978-3-3770-7

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Projektarbeit
Leistungskontrolle:	Bericht und Referat
Anteil Semesterwochenstunden:	5 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden sind in der Lage, eine Problemstellung selbstständig wissenschaftlich zu bearbeiten.

Bildung der Modulnote:

benoteter Bericht und Referat

Modulbeschreibung Bachelorarbeit

Schlüsselwörter: Abschlussarbeit, wissenschaftliches und ingenieurmäßiges Arbeiten, Projektarbeit

Zielgruppe: 7. Semester WKB **Modulnummer:** IT 105 7000

Arbeitsaufwand: 15 ECTS **450 h**
Davon
Kontaktzeit **40 h**
Selbststudium **340 h**
Prüfungsvorbereitung **70 h**

Unterrichtssprache: Deutsch oder Englisch
Modulverantwortung: Prof. Dr.-Ing. Reinhard Schmidt

Stand: 01.09.2019

Empfohlene Voraussetzungen:

- alle Prüfungsleistungen der ersten vier Semester müssen erfolgreich abgeschlossen sein
- abgeschlossenes Praxissemester
- fundierte Kenntnisse im eigenen Studienprofil

Modulziel – angestrebte Lernergebnisse:

Die Studierenden besitzen die Fähigkeit, sich in ingenieurmäßige Fragestellungen aus dem Bereich der Medieninformatik einzuarbeiten. Sie können wissenschaftliche und technische Weiterentwicklungen verstehen und auf Dauer verfolgen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Vorgehensweisen beim wissenschaftlichen Arbeiten

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- systematische Recherchen zu ingenieurwissenschaftlichen Fragestellungen durchführen

Übergreifende Kompetenzen

Die Studierenden erlangen

- detaillierte Einblicke und umfassende Erkenntnisse auf einem Teilgebiet der Technischen Informatik.

Inhalt:

In der Bachelorarbeit soll der Studierende zeigen, dass die während des Studiums erlernten Kenntnisse und erworbenen Fähigkeiten erfolgreich in die Praxis umgesetzt werden können. Dazu wird eine projektartige Aufgabe unter Einsatz von ingenieurmäßigen Methoden bearbeitet. Der betreuende Professor begleitet die Studierenden während der Bachelorarbeit und leitet sie zum wissenschaftlichen Arbeiten an. Die Arbeit schließt mit einer schriftlichen Ausarbeitung und einem Vortrag ab.

Literaturhinweise:

- Lutz Hering, Heike Hering: Technische Berichte, Vieweg, 2017, ISBN 978-3-8348-15-86-6.
- Bernd Heesen; Wissenschaftliches Arbeiten, Springer Verlag, 2014, ISBN 978-3-662-43346-1,
- Henning Lobin; Die wissenschaftliche Präsentation: Konzept – Visualisierung – Durchführung; Schönigh Verlag, 2012, ISBN 978-3-3770-7.
- Alfred Brink: Anfertigung wissenschaftlicher Arbeiten, Springer Gabler Verlag, 2013, ISBN 978-3-8349-4396-5
- Ragnar Müller, Jürgen Plieninger, Christian Rapp: Recherche 2.0, Springer Verlag, 2013, ISBN 978-3- 658- 02249-5

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Selbstständiges wissenschaftliches Arbeiten
Leistungskontrolle:	Bericht
Anteil Semesterwochenstunden:	12 SWS
Geschätzte studentische Arbeitszeit:	360 Stunden

Lernziele:

Die Studierenden beherrschen selbstständiges wissenschaftliches Arbeiten. Sie erwerben die Fähigkeit zum wissenschaftlichen und ingenieurmäßigen Arbeiten, sowohl eigenständig als auch im Projekt-Team.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Präsentation einer wissenschaftlichen Arbeit
Leistungskontrolle:	Referat (20 Minuten), Testat Teilnahme am IT-Kolloquium
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lernziele:

Die Studierenden können ihre eigene wissenschaftliche Arbeit präsentieren und überzeugend argumentieren.

Bildung der Modulnote:

gemittelte Note aus Bericht, Faktor 12 und Referat Faktor 3
unbenotetes Testat

Modulbeschreibung Wahlfachmodul

Schlüsselwörter: Vertiefung im eigenen Studienprofil

Zielgruppe: 7. Semester WKB **Modulnummer:** MD 7630

Arbeitsaufwand: 6 ECTS **180 h**
Davon
Kontaktzeit 120 h
Selbststudium 30 h
Prüfungsvorbereitung 30 h

Unterrichtssprache: Deutsch oder Englisch
Modulverantwortung: Prof. Dr.-Ing. Reinhard Schmidt

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Grundlegende Kenntnisse im eigenen Studienprofil Softwaretechnik.

Modulziel – angestrebte Lernergebnisse:

Die Studierenden erlangen eine wissenschaftliche und fachliche Vertiefung auf dem Gebiet der Softwaretechnik.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- aktuelle und industriennahe Techniken.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- aktuelle und industriennahe Techniken anwenden.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage

- aktuelle und industriennahe Techniken zu implementieren.

Inhalt:

Das Wahlfachmodul besteht aus Wahlpflichtfächern mit einem Umfang von insgesamt 6 SWS. Studierende wählen zur Vertiefung seines Studienprofils 3 Wahlfächer mit jeweils 2 SWS. Die zur Auswahl stehenden Wahlpflichtfächer werden zu Semesterbeginn öffentlich bekannt gegeben.

In den Wahlpflichtfächer werden aktuelle und industriennahe Techniken angeboten.

Literaturhinweise:

abhängig vom gewählten Wahlpflichtfach

Wird angeboten:

Wahlpflichtfächer werden jährlich angeboten.

Alle Wahlpflichtfächer sind im Modulhandbuch der Wahlpflichtfächer beschrieben.

Der Angebotsrhythmus ist ebenfalls im Modulhandbuch der Wahlpflichtfächer festgelegt.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	abhängig vom gewählten Wahlpflichtfach
Leistungskontrolle:	abhängig vom gewählten Wahlpflichtfach
Anteil Semesterwochenstunden:	3 x 2 SWS
Geschätzte studentische Arbeitszeit:	180 Stunden

Lernziele:

Die Studierenden verfügen über eine wissenschaftliche und fachliche Vertiefung im eigenen Studienprofil Softwaretechnik.

Bildung der Modulnote:

Mittelwert der Noten der Wahlpflichtfächer

Modulbeschreibung Wissenschaftliche Vertiefung

Schlüsselwörter: Eigenständiges Arbeiten in Entwicklung und Forschung

Zielgruppe: 7. Semester WKB **Modulnummer:** IT 105 7001

Arbeitsaufwand: 9 ECTS **270 h**
Davon
Kontaktzeit **20 h**
Selbststudium **210 h**
Prüfungsvorbereitung **40 h**

Unterrichtssprache: Deutsch oder Englisch
Modulverantwortung: Prof. Dr.-Ing. Reinhard Schmidt

Stand: 01.09.2019

Empfohlene Voraussetzungen:

Fundierte Kenntnisse im eigenen Studienprofil

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, sich in ingenieurmäßige Fragestellungen aus dem Bereich der Technischen Informatik einzuarbeiten, wissenschaftliche und technische Weiterentwicklungen zu verstehen und auf Dauer verfolgen zu können.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Vorgehensweisen beim wissenschaftlichen Arbeiten

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- systematische Recherchen zu ingenieurwissenschaftlichen Fragestellungen durchführen

Übergreifende Kompetenzen

Die Studierenden erlangen

- detaillierte Einblicke und umfassende Erkenntnisse auf einem Teilgebiet der Softwaretechnik.

Inhalt:

Recherche und Selbststudium im Umfeld der Bachelorarbeit

Literaturhinweise:

- Bernd Heesen; Wissenschaftliches Arbeiten, Springer Verlag, 2014, ISBN 978-3-662-43346-1,
- Alfred Brink: Anfertigung wissenschaftlicher Arbeiten, Springer Gabler Verlag, 2013, ISBN 978-3-8349-4396-5
- Ragnar Müller, Jürgen Plieninger, Christian Rapp: Recherche 2.0, Springer Verlag, 2013, ISBN 978-3- 658- 02249-5

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Recherche und Selbststudium
Leistungskontrolle:	Mündliche Prüfung (20 Minuten)
Anteil Semesterwochenstunden:	9 SWS
Geschätzte studentische Arbeitszeit:	270 Stunden

Lernziele:

Die Studierenden können aufgrund eigener Recherchen Problemstellungen der Softwaretechnik analysieren und eigenständig Problemlösungen finden und bewerten.

Bildung der Modulnote:

Mündliche Prüfung