

---

Fakultät für [Informatik und Ingenieurwissenschaften]

# **Modulhandbuch** **[Informatik]**

[Bachelor of Science]

# Inhalt

<b>Modulhandbuch   [Informatik], [Bachelor of Science]</b> .....	<b>3</b>
1 Studiengangbeschreibung .....	3
2 Absolvent*innenprofil .....	3
3 Handlungsfelder .....	4
4 Studienverlaufsplan .....	6
5 Alternativer Studienverlaufsplan .....	6
6 Module .....	7
6.1 [Einführungsprojekt in die Informatik] .....	7
6.2 [Algorithmen und Programmierung I] .....	8
6.3 [Einführung in Betriebssysteme und Rechnerarchitektur] .....	9
6.4 [Mathematik I] .....	10
6.5 [Theoretische Informatik I] .....	11
6.6 [Algorithmen und Programmierung II] .....	13
6.7 [Betriebssysteme und verteilte Systeme] .....	14
6.8 [BWL - Grundlagen] .....	15
6.9 [Mathematik II] .....	17
6.10 [Theoretische Informatik II] .....	18
6.11 [Algorithmik] .....	19
6.12 [Datenbanksysteme] .....	21
6.13 [Informations- und Prozessmanagement] .....	22
6.14 [Kommunikationstechnik und Netze] .....	23
6.15 [Projektmanagement] .....	25
6.16 [Softwaretechnik I] .....	26
6.17 [Informatik, Recht und Gesellschaft] .....	28
6.18 [Künstliche Intelligenz] .....	29
6.19 [Mensch-Computer-Interaktion] .....	31
6.20 [Paradigmen der Programmierung] .....	32
6.21 [Softwaretechnik II] .....	34
6.22 [Spezielle Gebiete der Datenbanken] .....	35
6.23 [Praxissemester] .....	37
6.24 [Wahlspezialisierung I] .....	38
6.25 [Wahlspezialisierung II] .....	39
6.26 [Praxisprojekt mit begleitendem Projektseminar] .....	40
6.27 [Bachelorarbeit] .....	41
6.28 [Bachelor Kolloquium] .....	42
7 Modulmatrix .....	44

# Modulhandbuch | [Informatik], [Bachelor of Science]

## 1 Studiengangbeschreibung

Die Digitalisierung ist eine Triebfeder zukunftsweisender Technologien. Sie verwischt zunehmend die Grenzen zwischen Analog und Digital. Seit einigen Jahren erfolgt daher eine Abkehr der IT-Industrie von einer industriellen, tayloristischen Arbeitskultur, die auf hohe Effizienz der (IT-)Arbeitsprozesse ausgerichtet ist, hin zu interdisziplinären, agilen und kreativ gestaltenden Arbeitsweisen. Informatikerinnen und Informatiker müssen dabei die Möglichkeiten der Informatik beherrschen und ihre heutigen Grenzen kennen. Sie gestalten, realisieren, betreiben und evaluieren in interdisziplinären Teams digitale Produkte und Prozesse. Dabei denken sie immer im Dreiklang Mensch, Aufgabe und Technik und handeln verantwortungsvoll, wenn sie Grenzen ausloten und verschieben. Information, Kooperation und Kommunikation sind im Studium Informatik ebenso wichtig wie Mathematik, formale Sprachen oder Algorithmen. Der stetige Fortschritt erfordert überdies nicht nur eine ständige Weiterentwicklung der Anwendungssoftware, sondern auch der eingesetzten Methoden, Techniken und Werkzeuge. Agile Vorgehensweisen, Flexibilität in Denken und Handeln und die Fähigkeit zum selbständigen lebenslangen Lernen sind somit weitere Kernkompetenzen von Informatikerinnen und Informatikern.

Der Studiengang Informatik bietet daher einen eng an »klassische« Informatik-Curricula angelehnten Kanon von IT-Kernfächern. Zusätzlich zu der bewährten Informatik-Grundausbildung bietet der Studiengang eine tiefgehende Spezialisierung in den Bereichen Algorithmik, Künstliche Intelligenz sowie modernen Methoden der Softwareentwicklung. Damit ist Informatik ein grundständiger, anwendungsorientierter Informatik-Bachelorstudiengang, der die Studierenden in besonderer Weise auf die technischen Anforderungen der Digitalisierung vorbereitet, dabei aber die gesellschaftlichen Folgen im Auge behält und die Studierenden dazu veranlasst, die Auswirkungen ihrer Arbeit sorgfältig und aufmerksam zu beachten. Algorithmen der Künstlichen Intelligenz dringen in (Entscheidungs- und Kompetenz-)Bereiche vor, welche bisher einzig und allein dem Menschen vorbehalten waren. Sie bieten großen Chancen für die Arbeitswelt, die Medizin, die Umwelt und das Zusammenleben der Menschen, aber auch Risiken, wenn sie kriminell missbraucht werden.

## 2 Absolvent\*innenprofil

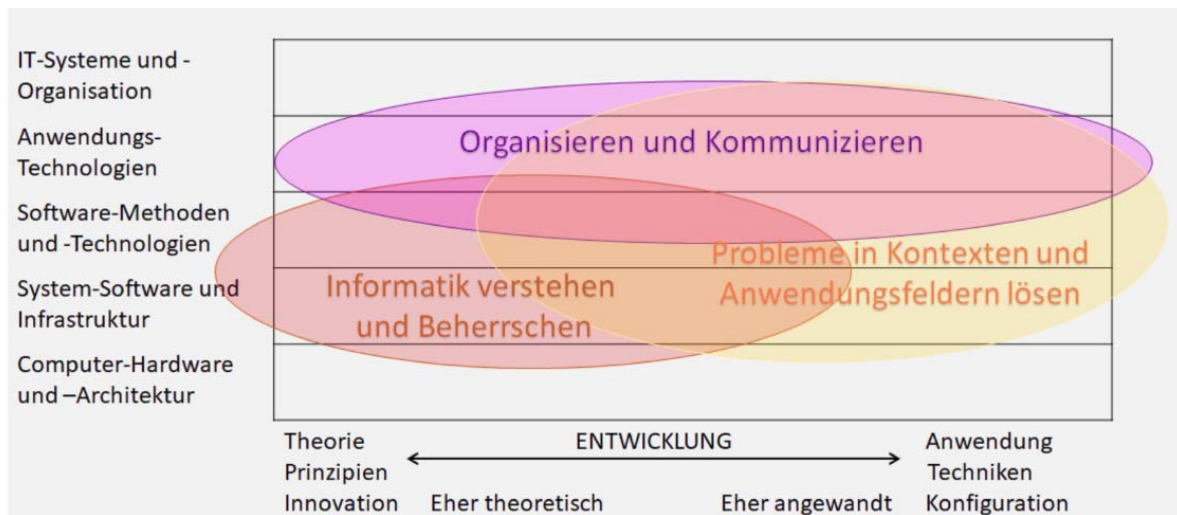
Der Studiengang Informatik bildet Informatiker\*innen aus, welche die Möglichkeiten der Informatik beherrschen und ihre heutigen Grenzen kennen. Sie modellieren, gestalten, realisieren, betreiben und evaluieren in interdisziplinären Teams digitale Produkte und technische Prozesse sowie menschliche Entscheidungsprozesse. Dabei denken sie im Dreiklang Mensch – Aufgabe – Technik und handeln verantwortungsvoll, wenn sie Grenzen ausloten und verschieben. Information, Kooperation und Kommunikation sind für die Absolvent\*innen des Studiengangs Informatik ebenso wichtig wie Mathematik, formale Sprachen oder Algorithmen. Der stetige Fortschritt erfordert überdies nicht nur eine ständige Weiterentwicklung der von ihnen entwickelten Anwendungssoftware, sondern auch ihrer dabei eingesetzten Methoden, Techniken und Werkzeuge. Agile Vorgehensweisen, Flexibilität in Denken und Handeln und die Fähigkeit zum selbstständigen lebenslangen Lernen sind somit weitere Kernkompetenzen von Absolvent\*innen des Studiengangs Informatik.

### Kompetenzcluster des Studiengangs Informatik:

- Strukturieren und Programmieren
- Softwarequalität und Faktor Mensch
- Innovation
- Produktentwicklung
- Kommunikation
- Team- und projektorientiertes Arbeiten
- Wissenschaftliches und eigenverantwortliches Arbeiten

### 3 Handlungsfelder

Die verschiedenen Tätigkeiten, für deren Ausübung die Absolvent\*innen im Studiengang Informatik die aufgeführten Kompetenzen bilden, lassen sich in drei **Handlungsfelder** unterteilen. Die Handlungsfelder werden bestimmten Bereichen konkreter Kompetenzen zugeordnet, die wiederum von den Lernzielen und Inhalten der Module des Curriculums abgedeckt werden.



(Bild angelehnt an Fig 2.7 p21, <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2005-march06final.pdf>, letzter Zugriff: 12.01.2020)

#### Informatik verstehen und beherrschen

Für das berufliche Handlungsfeld »Informatik Beherrschen« haben die Absolventen ein tiefes, umfassendes Verstehen und Wissen über die grundlegenden Bereiche, Wissens- und Fachgebiete (Body of Knowledge) der Informatik entwickelt. Hierzu gehören:

- die Lösung von Entwicklungsaufgaben mittels grundlegender formaler, algorithmischer, mathematischer Methoden, Techniken und Verfahren,
- die Erkennung und Nutzung wiederkehrender Themen wie Abstraktion, Komplexität, evolutionäre Veränderungen, Modellierung, etc. und allgemeiner Prinzipien wie z.B. gemeinsame Ressourcen-Nutzung und Nebenläufigkeit,
- das Erkennen und das Tiefe Verständnis davon, welche Themen und Prinzipien eine gemeinsame und breite Anwendung in den verschiedenen Teilgebieten der Informatik haben,
- die Anwendung von fachlich vertiefenden Wissens- und Handlungsbereichen wie Softwaretechnik, Datenbanken, Algorithmik, Künstliche Intelligenz, IT-Security, Informationsmanagement, Mensch-Computer Interaktion.

Für das berufliche Handlungsfeld „Informatik verstehen und beherrschen“ bilden die Studierenden im Studiengang Kompetenzen, um realweltliche (Teil-)Problemstellungen mit grundlegenden Informatik-Techniken und Verfahren lösen zu können.

### **Probleme in Kontexten und Anwendungsfeldern lösen**

Im Bereich der kontextbezogenen Problemlösung modellieren und abstrahieren Absolvent\*innen die fachlichen und technischen Aspekte von Software-Systemen. Das Verstehen der Probleme und Aufgaben muss z.B. über Implementierungsdetails hinausgehen. Dazu muss ein Verständnis für die Struktur von Computersystemen und die entsprechenden Prozesse, die an der Analyse und Konstruktion beteiligt sind, entwickelt werden. Die Absolventen verstehen, dass Informatik mit vielen verschiedenen Fach-Domänen und Anwendungsbereichen verknüpft ist. Daher ist Fähigkeit zur Kommunikation mit Experten aus anderen Fachdisziplinen für Informatik-Absolventen besonders wichtig, um die für Problemlösungen erforderlichen Domänenkenntnisse zu erlangen. Zentrale Fragen sind: "In welchem Kontext werden Computersysteme eingesetzt?" und "Welche Interaktionen mit Menschen und der physischen Welt finden statt? Aktivitäten der Absolvent\*innen in diesem Handlungsfeld sind beispielsweise:

- das Einnehmen von Perspektiven unterschiedlicher Detail- und Abstraktionsebenen,
- das Beachten und Wertschätzen Domänen-spezifischen Wissens das Verstehen und kontextbezogene Gestalten des Software-Lebenszyklus mit Analyse-, Entwurfs-, Realisierungs-, Evaluierungstätigkeiten,
- der Einsatz dedizierter Projektvorgehensweisen inklusive agiler Vorgehensweisen bei Green- und Brownfield-Vorhaben,
- die Planung und Evaluierung von Systemen auf der Grundlage von Funktionalität, Usability, User Experience und Leistung,
- das eigentliche Programmieren (schreiben und debuggen von Quellcode),
- das Entwerfen von Software-Architekturen,
- das Entwickeln von Services im Backend und von Frontends,
- das Testen von Programmcode, Komponenten und Systemen,
- das Absichern von Software-Systemen,
- das Bereitstellen und Betreiben von Software-Systemen.

Die Absolvent\*innen erwerben für dieses Handlungsfeld Kompetenzen, um Software-Systeme vom Erheben der Anforderungen über deren Validierung z.B. mit low-fidelity-Prototypen bis hin zur Realisierung und Bereitstellung sowie dem Betrieb entwickeln zu können – Sie können also reale Systeme „bauen“.

### **Organisieren und Kommunizieren**

Die Fähigkeit und Bereitschaft zur Teamarbeit sind eine grundlegende Eigenschaft, welche die Absolventen während ihres Studiums erlernen und trainieren sollen. Ebenso gehören das Projektmanagement und das Gestalten eigener Lernwege, das Zeitmanagement und die Priorisierung von zu erledigenden Aufgaben zu den grundlegenden Fähigkeiten im Handlungsbereich querschnittliche Qualifikationen. Zu den Aktivitäten der Absolvent\*innen in diesem Handlungsfeld zählen z.B.:

- das selbstorganisierte Arbeiten in agilen Entwicklungsteams,
- die Erklärung, Begründung und Präsentation eigener Lösungen,
- die Berücksichtigung von sozialen, rechtlichen, ethischen und kulturellen Aspekten,
- das Treffen von Entscheidungen und die Auswahl von Handlungsalternativen im Bewusstsein möglicher Folgen und der damit einhergehenden individuellen und kollektive Verantwortung,
- die Weiterentwicklung und Anpassung der eigenen Fähigkeiten während des gesamten (Berufs-)Lebens.

Absolvent\*innen bilden für dieses Handlungsfeld Kompetenzen aus, um kleinen bis großen Unternehmen und Organisationen und unterschiedlichen Führungs- und Arbeitsstilen wirken zu können.

### 4 Studienverlaufsplan

Studienabschnitte Informatik			Leistungspunkte & Semesterzuordnung							
Semester	Module	PV	Σ	1	2	3	4	5	6	7
	<b>Grundlagen</b>		<b>60</b>							
1	Einführungsprojekt in die Informatik	-	5	5						
	Einführung in Betriebssysteme und Rechnerarchitektur	-	5	5						
	Mathematik I	PV	7	7						
	Algorithmen und Programmierung I	PV	8	8						
	Theoretische Informatik I	-	5	5						
2	Mathematik II	PV	8		8					
	Algorithmen und Programmierung II	PV	7		7					
	Theoretische Informatik II	-	5		5					
	Betriebssysteme und verteilte Systeme	PV	5		5					
	BWL	-	5		5					
	<b>Vertiefung</b>		<b>60</b>							
3	Algorithmik	PV	5			5				
	Projektmanagement	-	5			5				
	Kommunikationstechnik und Netze	PV	5			5				
	Softwaretechnik I	PV	5			5				
	Datenbanksysteme	PV	5			5				
	Informations- und Prozessmanagement	-	5			5				
	Spezielle Gebiete der Datenbanken	PV	5				5			
4	Paradigmen der Programmierung	PV	5				5			
	Softwaretechnik II	PV	5				5			
	Informatik, Recht und Gesellschaft	-	5				5			
	Künstliche Intelligenz	PV	5				5			
	Mensch-Computer-Interaktion	PV	5				5			
Praxissemester		<b>30</b>								
5	Praxissemester	-	30					30		
	<b>Spezialisierung</b>		<b>60</b>							
6	Wahlspezialisierung I		15						15	
	Wahlspezialisierung II		15						15	
7	Praxisprojekt mit begleitendem Projektseminar	-	15							15
	Bachelor Arbeit	-	12							12
	Bachelor Kolloquium	-	3							3
<b>Σ: Studiengang Informatik</b>			<b>Σ</b>	<b>210</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>	<b>30</b>

### 5 Alternativer Studienverlaufsplan

Studienabschnitte Informatik (Teilzeit)			Leistungspunkte & Semesterzuordnung										
Semester	Module	PV		1	2	3	4	5	6	7	8	9	10
	<b>Grundlagen</b>		<b>40</b>										
1	Einführungsprojekt in die Informatik	-	5	5									
	Algorithmen und Programmierung I	PV	8	8									
	Mathematik I	PV	7	7									
2	Algorithmen und Programmierung II	PV	7		7								
	Mathematik II	PV	8		8								
	BWL	-	5		5								
	<b>Grundlagen/Vertiefung</b>		<b>80</b>										
3	Informations- und Prozessmanagement	-	5			5							
	Einführung in Betriebssysteme und Rechnerarchitektur	-	5			5							
	Theoretische Informatik I	-	5			5							
	Softwaretechnik I	PV	5			5							
4	Betriebssysteme und verteilte Systeme	PV	5				5						
	Softwaretechnik II	PV	5				5						
	Theoretische Informatik II	-	5				5						
	Informatik, Recht und Gesellschaft	-	5				5						
5	Kommunikationstechnik und Netze	PV	5					5					
	Datenbanksysteme	PV	5					5					
	Projektmanagement	-	5					5					
	Algorithmik	PV	5					5					
6	Künstliche Intelligenz	PV	5						5				
	Paradigmen der Programmierung	PV	5							5			
	Mensch-Computer-Interaktion	PV	5								5		
	Spezielle Gebiete der Datenbanken	PV	5									5	
7	Praxissemester		<b>30</b>										
	Praxissemester	-	30								30		
	<b>Spezialisierung</b>		<b>30</b>										
8	Wahlspezialisierung I		15									15	
9	Wahlspezialisierung II		15										15
	<b>Abschlusssemester</b>		<b>30</b>										
10	Praxisprojekt mit begleitendem Projektseminar	-	15										15
	Bachelor Arbeit	-	12										12
	Bachelor Kolloquium	-	3										3
				<b>210</b>	<b>20</b>	<b>20</b>	<b>20</b>	<b>20</b>	<b>20</b>	<b>20</b>	<b>20</b>	<b>30</b>	<b>15</b>

## 6 Module

### 6.1 [Einführungsprojekt in die Informatik]

Modulnummer:	EI
Modulbezeichnung:	Einführungsprojekt in die Informatik
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	1
Häufigkeit des Angebots:	Jedes Wintersemester
Modulverantwortliche*r:	Prof. Dr. Heinrich Klocke
Dozierende:	Heinrich Klocke, Sven Kullack B.Sc., Dipl. Inf. Alex Maier M.Sc., Pascal Schönthier M.Sc.,
Learning Outcome:	(WAS) Dieses Modul versetzt die Studierenden in die Lage, die späteren Inhalte des Studiengangs im Gesamtkontext des Software- Lebenszyklus einzuordnen. (WOMIT) Dies geschieht im Rahmen von Projekten, in denen Studierende erste lauffähige Software Produkte umsetzen. (WOZU) Studierende identifizieren mögliche Handlungsfelder für einen Informatik-Beruf und erhalten einen Gesamtüberblick über den Studienverlauf sowie die Zusammenhänge.
Modulinhalte:	<p>In diesem Modul erhalten die Studierenden eine Einführung in ihren Studiengang. Die Veranstaltung gliedert sich in eine einführende Projektwoche, in der eine erste lauffähige Software-anwendung implementiert wird. Anschließend folgt eine Vorlesungsreihe mit projektorientierten Anwendung der vermittelten Themen:</p> <ul style="list-style-type: none"> <li>• Einführung in studiumsbegleitende Lernplattformen und Hochschulinfrastruktur</li> <li>• Umgang mit Tools, die im weiteren Verlauf des Studiums wichtig sind: <ul style="list-style-type: none"> <li>• Source Code Versionsverwaltung mit git</li> <li>• Continuous Integration/Delivery/Deployment (Jenkins/Bamboo)</li> <li>• Projektmanagement Tools (JIRA/Redmine/Trello)</li> <li>• Collaboration Tools (WIKI / Confluence)</li> <li>• Chat(Ops) (Slack)</li> <li>• Test-Frameworks (J-Unit/gtest?/Integrationstest)</li> </ul> </li> <li>• Methoden die in der Software-Entwicklung und in der Informatik relevant sind <ul style="list-style-type: none"> <li>• Test-Driven-Development (TDD)</li> <li>• Agile Vorgehensmodelle (Scrum oder generischere wie eXtreme Programming)</li> <li>• DevOps (die Kunst seine eigene Software zu betreiben)</li> <li>• Code Review</li> <li>• SCV Branch-Modelle (Feature Branch)</li> </ul> </li> <li>• Wissenschaftliches Arbeiten <ul style="list-style-type: none"> <li>• Wissenschaftliche Ausarbeitungen erstellen mit latex</li> <li>• Literaturverwaltung (bibtex?)</li> <li>• Thesenbildung</li> <li>• Aufsetzen von Experimenten? (bessere Formulierung TBD)</li> </ul> </li> <li>• Abschließend werden die Studierenden ein selbst ausgesuchtes Software-Projekt in einer Blockveranstaltung umsetzen und gestalten, dieses Projekt wird im Rahmen eines Hackatons umgesetzt.</li> </ul>
Lehr- und Lernmethoden:	4 SWS Angeleitete Projektarbeit
Prüfungsformen:	Klausur oder Fallbeispiel mit Präsentation. Die Prüfungsform wird beim ersten Termin mitgeteilt.
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	150 h
Präsenzzeit:	60 h / 4 SWS
Selbststudium:	90 h
Empfohlene Voraussetzungen:	keine über die Zulassungsbedingungen hinausgehenden Voraussetzungen

Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<b>Versionsverwaltung mit Git</b> (mitp Professional) ISBN: 3958452264
Verwendung des Moduls in weiteren Studiengängen:	keine
Besonderheiten:	keine
Letzte Aktualisierung:	20.10.2020

## 6.2 [Algorithmen und Programmierung I]

Modulnummer:	AP1
Modulbezeichnung:	Algorithmen und Programmierung I
Art des Moduls:	Pflicht
ECTS credits:	8
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	1
Häufigkeit des Angebots:	Jedes Wintersemester
Modulverantwortliche*r:	Prof. Dr. Frank Victor
Dozierende:	Prof. Dr. Frank Victor
Learning Outcome:	Die Studierenden <ul style="list-style-type: none"> <li>• (WAS) verstehen die Prinzipien der prozeduralen Programmierung und der Objektorientierung, (WOMIT) indem sie die in der Lehrveranstaltung vorgestellten Konzepte auf strukturierte und unstrukturierte Problemstellungen anwenden, (WOZU) um diese in der Softwareentwicklung einzusetzen.</li> <li>• (WAS) können die Anforderungen an Programmsysteme analysieren, (WOMIT) indem sie die in der Vorlesung und Übung behandelten Methoden modellbasiert adaptieren, (WOZU) um diese geeignet in Software-Lösungen umzusetzen.</li> <li>• (WAS) können Systementwürfe evaluieren und bewerten, (WOMIT) indem sie die besprochenen Komplexitäts- und Qualitätskriterien anwenden, (WOZU) um die Qualität der entwickelten Software zu erhöhen.</li> <li>• (WAS) erlernen und trainieren, algorithmische Entwurfsmuster zu erkennen und anzuwenden, (WOMIT) indem sie die in der Veranstaltung besprochenen Standardalgorithmen verstehen, modifizieren und anpassen, (WOZU) um diese in komplexe Programmsysteme zu integrieren.</li> </ul>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Prozedurale Programmierung am Beispiel der Programmiersprache C.</li> <li>• Objektorientierte Programmierung am Beispiel von Java.</li> <li>• Kontroll- und Datenstrukturen.</li> <li>• Modularisierungskonzepte.</li> <li>• Typkonzepte.</li> <li>• Grundmuster der objektorientierten Programmierung.</li> <li>• Elementare Algorithmen und Aufwandsschätzung.</li> <li>• Entwicklungsumgebungen.</li> </ul>
Lehr- und Lernmethoden:	<ul style="list-style-type: none"> <li>• Vorlesung mit Folien und Programmbeispielen zum Download</li> <li>• Übungen in Teamarbeit mit vorbereitenden Aufgaben zum Praktikum und zur Vertiefung der Inhalte der Vorlesung</li> <li>• Praktikum mit individueller Abnahme der Programmieraufgaben</li> <li>• Beratungen in kleinen Gruppen zu den Praktikaaufgaben und zu Verständnisfragen</li> </ul>
Prüfungsformen:	Klausur sowie erfolgreiche Teilnahme am Praktikum
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	240 h



Präsenzzeit:	110 h (3 SWS Vorlesung, 1 SWS Übung, 2 SWS Praktikum)
Selbststudium:	130 h
Empfohlene Voraussetzungen:	keine über die Zulassungsbedingungen hinausgehenden Voraussetzungen
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Vorlesungsunterlagen: Foliensammlung, Übungen mit Lösungen, Übungsklausuren mit Lösungen</li> <li>• Fachliteratur: Diverse C-Bücher, u.a.: Kernighan, B.W., Ritchie, D.M.: „Programmieren in C“</li> <li>• Diverse Java-Bücher, u.a.: Bishop, J.: „Java Lernen“</li> <li>• Sedgewick, R.: „Algorithmen in Java“</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Bachelor: IT-Management (Grundlagen, 1. Sem.) Bachelor: Wirtschaftsinformatik (Grundlagen, 1. Sem.) Bachelor: Medieninformatik (Grundlagen, 1. Sem.)
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

### 6.3 [Einführung in Betriebssysteme und Rechnerarchitektur]

Modulnummer:	EBR
Modulbezeichnung:	Einführung in Betriebssysteme und Rechnerarchitektur
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	1
Häufigkeit des Angebots:	Jedes Wintersemester
Modulverantwortliche*r:	Prof. Dr. Stefan Karsch
Dozierende:	Prof. Dr. Stefan Karsch
Learning Outcome:	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>• (WAS) können Breite und Tiefe der Informatik als Wissenschaftsdisziplin einschätzen, (WOMIT) in dem sie exemplarische Fachfragestellungen der Informatik einordnen, (WOZU) um die Bedeutung der von Ihnen gewählten Wissenschaftsdisziplin zu erkennen und ggf. später eine fundiertere Spezialisierung auszuwählen,</li> <li>• (WAS) kennen und verstehen Fachbegriffe der Rechnerarchitektur, (WOMIT) in dem Sie diese terminologisch, ausgehend von der von-Neumann'schen Basisarchitektur einordnen, (WOZU) um später Fachgespräche zur Systemgestaltung oder Programmierung fachgerecht zu führen und Fachdokumente professionell anzufertigen,</li> <li>• (WAS) können die Grundelemente von Rechnerarchitekturen einordnen, (WOMIT) in dem sie sich die Wirkung auf die Leistungsfähigkeit von Rechnern mit Beispielen in Vorlesung und Übung veranschaulichen, (WOZU) um aktuelle und kommende technologische Innovationen bewerten zu können,</li> <li>• (WAS) kennen und verstehen Grundlagen von Betriebssystemen, (WOMIT) in dem sie die in Vorlesung und Übung vorstellten Konzepte auf konkrete Beispielarchitekturen anwenden, (WOZU) um diese dann einordnen zu können,</li> <li>• (WAS) kennen und verstehen die grundlegenden Komponenten von Betriebssystemen und ihr Zusammenwirken, (WOMIT) in dem sie sich diese Elemente im Verlauf der Vorlesung schrittweise erschließen, (WOZU) um später mit realen Systemen kompetent umzugehen (bspw. bei der Bedienung, bei der Programmierung und beim Betrieb der Systeme).</li> <li>• (WAS) kennen und verstehen die Bedeutung von Programmierschnittstellen und Systembibliotheken, (WOMIT) in dem sie sich deren Zusammenwirken mit anderen</li> </ul>

	Systemkomponenten schrittweise veranschaulichen, (WOZU) um später Systemeigenschaften und Systemschnittstellen bei der Programmierung einzusetzen.
Modulinhalte:	<ul style="list-style-type: none"> <li>• Grundlagen: Was ist Informatik? Geschichte der IT, Zahlen – und Zeichendarstellung in Rechnersystemen</li> <li>• Grundlagen der Rechnerarchitektur: Von Neumann Architektur, Speicherhierarchie, physikalischer Aufbau von magnetischen und elektronischen Speichermedien, physikalischer Aufbau optischer Speichermedien, Busse und Schnittstellen, Beispielarchitekturen</li> <li>• Grundlagen von Betriebssystemen: Schichtenmodell, Betriebsarten, Programmausführung, Prozesse und Scheduling, Beispiel: Der BSD-Unix Scheduler, Interrupts, Speicherverwaltung: demand paging, working set, Auslagerungsverfahren, Beispiel: demand paging unter BSD-Unix, Dateisysteme, Beispiele: Unix inodes und MSDOS FAT, Rechteverwaltung, Netzwerkbetriebssysteme</li> <li>• Im Mittelpunkt der Veranstaltung steht die Vermittlung von Basiskonzepten und Grundlagen, die sich auf die Benutzung von Betriebssystemen beziehen. Das Design von Betriebssystemen und die konkrete Systemprogrammierung werden im Modul Betriebssysteme behandelt, das auf den Grundlagen des Faches EBR aufbaut.</li> </ul>
Lehr- und Lernmethoden:	<ul style="list-style-type: none"> <li>• Vorlesung mit Folien und Grafiken zum Download</li> <li>• Übungen in Teamarbeit mit Aufgaben zur Vertiefung der Inhalte der Vorlesung</li> <li>• Beratungen im Rahmen von Übungsterminen kleinen Gruppen zu den Übungsaufgaben und zu Verständnisfragen</li> </ul>
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	72 h (2 SWS Vorlesung, 2 SWS Übung)
Selbststudium:	78 h
Empfohlene Voraussetzungen:	keine über die Zulassungsbedingungen hinausgehenden Voraussetzungen
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Vorlesungsunterlagen: kommentierte Foliensammlung</li> <li>• Tanenbaum: „Rechnerarchitektur“</li> <li>• Tanenbaum: „Modern Operating Systems“</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Bachelor: IT-Management (Grundlagen, 1. Sem.) Bachelor: Informatik (Grundlagen, 1. Sem.) Bachelor: Medieninformatik (Grundlagen, 1. Sem.) Bachelor: Wirtschaftsinformatik (Grundlagen, 1. Sem.)
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.4 [Mathematik I]

Modulnummer:	MA1
Modulbezeichnung:	Mathematik I
Art des Moduls:	Pflicht
ECTS credits:	7
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	1
Häufigkeit des Angebots:	Jedes Wintersemester

Modulverantwortliche*r:	Prof. Dr. Wolfgang Konen
Dozierende:	Prof. Dr. Wolfgang Konen
Learning Outcome:	<ul style="list-style-type: none"> <li>• (WAS) Ziel des Kurses ist eine Einführung in die grundlegenden Begriffe, Methoden Techniken der Mathematik für die Informatik anhand der ausgewählten Teilgebiete.</li> <li>• (WOMIT) Die Studierenden erwerben die Fähigkeiten zur Analyse realer oder geplanter Systeme, indem sie praktische Aufgabenstellungen aus dem Informatik-Umfeld in mathematische Strukturen abstrahieren und lernen.</li> <li>• (WOZU) Die Studierenden erkennen die Anwendungsbezüge der Mathematik für die Informatik, z.B. die Bedeutung funktionaler Beziehungen für kontinuierliche Zusammenhänge, die lineare Algebra als Grundlage der grafischen Datenverarbeitung und die Analysis zur Verarbeitung von Signalen und zur Lösung von mathematischen Modellen.</li> </ul>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Grundlagen</li> <li>• Logik</li> <li>• Folgen und Grenzwerte</li> <li>• Analysis (einer Veränderlichen)</li> <li>• Lineare Algebra</li> </ul>
Lehr- und Lernmethoden:	<ul style="list-style-type: none"> <li>• Vorlesung mit Skript und Tabletmitschrieb zum Download</li> <li>• Übungen mit vorbereitenden Aufgaben, die von Studenten vorgestellt und gemeinsam diskutiert werden, zur Vertiefung der Inhalte der Vorlesung</li> <li>• Praktikum mit individueller Abnahme der mathematischen Maple- und Mathweb-Aufgaben</li> <li>• fallweise Tutorien mit studentischen Tutor*innen zur Klärung Verständnisfragen</li> </ul>
Prüfungsformen:	Klausur sowie erfolgreiche Teilnahme am Praktikum
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	210h
Präsenzzeit:	100 h (3 SWS V + 2 SWS Ü + 1 SWS P)
Selbststudium:	110h
Empfohlene Voraussetzungen:	keine über die Zulassungsbedingungen hinausgehenden Voraussetzungen
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<p>Skript unter <a href="http://www.gm.fh-koeln.de/~konen/Mathe1-WS">www.gm.fh-koeln.de/~konen/Mathe1-WS</a>, dort auch weitere Literatur, z. B.:</p> <ul style="list-style-type: none"> <li>• <b>Teschl, Gerald und Teschl, Susanne:</b> "Mathematik für Informatiker", 4. Auflage, Springer Verlag, 2013</li> <li>• <b>Hartmann, Peter:</b> "Mathematik für Informatiker – Ein praxisbezogenes Lehrbuch", 7. Auflage, Vieweg Verlag, 2020</li> <li>• <b>Papula, Lothar:</b> "Mathematik für Ingenieure und Naturwissenschaftler" Vieweg Verlag, 2012</li> <li>• <b>Knorrenschild, Michael,</b> "Vorkurs Mathematik", 4. Auflage, Hanser-Verlag, 2013</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	<p>Bachelor: IT-Management (Grundlagen, 1. Sem.)          Bachelor: Informatik (Grundlagen, 1. Sem.)          Bachelor: Medieninformatik (Grundlagen, 1. Sem.)</p>
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.5 [Theoretische Informatik I]

Modulnummer:	T11
Modulbezeichnung:	Theoretische Informatik I
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch

Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	1
Häufigkeit des Angebots:	Jedes Wintersemester
Modulverantwortliche*r:	Prof. Dr. Martin Eisemann
Dozierende:	Prof. Dr. Martin Eisemann
Learning Outcome:	<ul style="list-style-type: none"> <li>(WAS) Grundsätzliches Ziel des Kurses ist eine Einführung in die Begriffe, Methoden, Modelle und Arbeitsweise der Theoretischen Informatik anhand der ausgewählten Teilgebiete. (WOMIT) Dabei lernen die Studierenden Probleme und Sachverhalte zu abstrahieren und zu modellieren (etwa logische und algebraische Kalküle, graphentheoretische Notationen, formale Sprachen und Automaten sowie spezielle Kalküle wie Petri-Netze). (WOZU) Die Studierenden erwerben fundierte Kenntnisse der grundlegenden Themengebiete und eine wesentliche Basis und Vorbereitung für Veranstaltungen in höheren Semestern des Studiums.</li> <li>(WAS) In verschiedenen Grundlagengebieten der Informatik lernen die Studierenden Verfahrensweisen kennen, um den algorithmischen Kern eines Problems zu identifizieren und können passende Algorithmen entwerfen (Automaten, Turing Maschinen, Logik). (WOMIT) Aufgaben zu den Lehrinhalten (s.u.) werden in kleinen Gruppen (Teamarbeit) selbständig gelöst. Die Lösungen sollen in den Übungsstunden vorgetragen und der Lösungsweg den Kommilitonen hierbei erläutert werden. (WOZU) Dabei können sie bekannte Problemstellungen im Anwendungskontext erkennen und sind mit den zugehörigen Lösungsmustern vertraut (Modellierung mittels Automaten, Petri-Netzen, Boolescher Algebra, etc.).</li> </ul>
Modulinhalte:	<ul style="list-style-type: none"> <li>Mengen</li> <li>Relationen</li> <li>Graphen</li> <li>Zahlensysteme</li> <li>Zahlendarstellung</li> <li>Numerische Aspekte</li> <li>Codierung, Informationstheorie</li> <li>Boolesche Algebra</li> <li>Schaltnetze und Schaltwerke</li> <li>Aussagenlogik</li> <li>Prädikatenlogik</li> </ul>
Lehr- und Lernmethoden:	Vorlesung mit Übungen - 4 SWS: Vorlesung 2 SWS; Übung 2 SWS
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	Einfache Kenntnisse der naiven Mengenlehre, wie sie in der Schule vermittelt und bei der mathematischen Begriffsbildung verwendet werden, werden empfohlen. Keine über die Zulassungsvoraussetzungen hinausgehenden Voraussetzungen.
Zwingende Voraussetzungen:	Keine
<ul style="list-style-type: none"> <li>Empfohlene Literatur:</li> </ul>	<ul style="list-style-type: none"> <li>Hoffmann, D. (2011): Theoretische Informatik, 2. Auflage</li> <li>Hedtstück, U. (2004): Einführung in die Theoretische Informatik. Oldenbourg, München.</li> <li>Kelly, J. (2003): Logik. Pearson Studium, München.</li> <li>Ehrig, H. et al. (1999): Mathematisch-strukturelle Grundlagen der Informatik. Springer, Heidelberg.</li> <li>Beuth, K. (1992): Digitaltechnik. 9.Aufl. Vogel, Würzburg.</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Bachelor: Medieninformatik (Grundlagen, 1. Sem.)
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.6 [Algorithmen und Programmierung II]

Modulnummer:	AP2
Modulbezeichnung:	Algorithmen und Programmierung II
Art des Moduls:	Pflicht
ECTS credits:	7
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	2
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Christian Kohls
Dozierende:	Prof. Dr. Christian Kohls
Learning Outcome:	<p>Die Studierende sollen</p> <ul style="list-style-type: none"> <li>• (WAS) die grundlegenden Prinzipien, Strukturen und Syntaxelemente der objekt-orientierten Programmierung verstehen und anwenden können, (WOMIT) indem sie die Programmiersprache Kotlin einsetzen, (WOZU) um eigene Algorithmen umsetzen zu können,</li> <li>• (WAS) Standardalgorithmen (z.B. Suchen, Sortieren) verstehen und anwenden, (WOMIT) indem Sie Pseudocode in lauffähigen Code übersetzen, (WOZU) um deren Eigenschaften (z.B. Laufzeitverhalten) und Funktionsweise einordnen zu können,</li> <li>• (WAS) eigene objektorientierte Datenstrukturen entwickeln und die Komposition komplexer Objektstrukturen beherrschen, (WOMIT) indem sie Geschäfts- und Anwendungsfelder der realen Welt als Software modellieren und verschiedene abstrakte Datentypen (z.B. Listen) und unterschiedliche konkrete Implementierungen (z.B. Verkettete Listen) umsetzen, (WOZU) um die Planung und den Entwurf einfacher Softwarearchitekturen durchzuführen und Systeme zu modellieren</li> <li>• (WAS) gut strukturierten, dokumentierten und wartbaren Code entwerfen können, (WOMIT) indem Prinzipien des Clean Coding, Entwurfsmuster und Teststrategien angewandt werden, (WOZU) um robuste und sichere Software zu entwickeln, eigene Software in einer objektorientierten Programmiersprache planen, entwickeln, umsetzen und testen könne</li> </ul> <p>Hierzu werden die Programmiersprache Kotlin, grundlegende Bibliotheken (Java, Android) und eine integrierte Entwicklungsumgebung eingesetzt.</p>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Arbeiten mit integrierter Entwicklungsumgebung</li> <li>• Dynamische vs. Statische Typisierung</li> <li>• Objekte und Abstraktion</li> <li>• Einfache Klassen</li> <li>• Datenkapselung</li> <li>• Konstruktoren</li> <li>• Veränderbare und unveränderbare Listen</li> <li>• when-Anweisungen und Ausdrücke</li> <li>• Vererbung und Typen</li> <li>• Klassenhierarchien</li> <li>• Typkompatibilität, Upcast, Downcast und Smart Cast</li> <li>• Polymorphie und Dynamische Bindung</li> <li>• Objektkomposition und Objekttaggregation</li> <li>• Parametrisierte Datentypen (Generics)</li> <li>• Begleit-Objekte für Klassen</li> <li>• Abstrakte Klassen und Schnittstellen</li> <li>• Nullfähige Typen</li> <li>• Abstrakte Datentypen in der Programmierung</li> <li>• Einfache und doppelt verkettete Liste</li> <li>• Exceptions</li> <li>• Clean Code</li> <li>• Testfälle schreiben</li> <li>• Laufzeiteffizienz und O-Notation</li> <li>• Entwurfsmuster (Iterator, Observer, Singleton, Strategie u.a.)</li> </ul>

	<ul style="list-style-type: none"> <li>• Verschachtelte, innere, statische und lokale Klassen</li> <li>• Sortieralgorithmen, u.a. Quicksort</li> <li>• Listen-Funktionen und Lambda-Ausdrücke</li> <li>• Stacks</li> <li>• Warteschlange</li> <li>• Assoziative Speicher</li> <li>• Nicht-lineare Datenstrukturen</li> <li>• Binärbäume</li> <li>• Verarbeitung durch Funktionen höherer Ordnung</li> <li>• Vergleich mit anderen Programmierparadigmen</li> </ul>
Lehr- und Lernmethoden:	<ul style="list-style-type: none"> <li>• Vorlesung mit interaktiven Phasen, Präsentationen und Live-Coding</li> <li>• Übung</li> <li>• Praktikum</li> <li>• Selbststudium mit bereitgestellten Screencasts, einem umfassenden Skript sowie Fachliteratur</li> <li>• 6 SWS: Vorlesung 3 SWS; Übung 1 SWS; Praktikum 2 SWS</li> </ul>
Prüfungsformen:	60 minütige Klausur sowie erfolgreiche Teilnahme am Praktikum
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	210 h
Präsenzzeit:	108 h
Selbststudium:	102 h
Empfohlene Voraussetzungen:	Erfolgreiche Teilnahme an AP1 wird empfohlen
Zwingende Voraussetzungen:	keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Vorlesungsunterlagen: Foliensammlung, ausformuliertes Skript, Beispiellösungen, Screencasts</li> <li>• Kohls, C., Dobrynin, A., Leonard, F. (2020) Programmieren Lernen mit Kotlin. München: Hanser Verlag.</li> <li>• Dmitry Jemerov &amp; Svetlana Isakova (2017). Kotlin in Action. Manning Publications.</li> <li>• Dawn Griffiths, David Griffiths &amp; Jørgen W. Lang (2019). Kotlin von Kopf bis Fuß: Eine Einführung in die Kotlin-Programmierung. O'Reilly.</li> <li>• Thomas Theis (2019). Einstieg in Kotlin: Apps entwickeln mit Android Studio. Keine Vorkenntnisse erforderlich, ideal für Kotlin-Einsteiger und Java-Umsteiger. Rheinwerk-Verlag</li> <li>• Karl Szwillus (2019). Kotlin: Einstieg und Praxis. mitp Professional.</li> <li>• Online-Referenz und Tutorials: <a href="https://kotlinlang.org/docs/reference/">https://kotlinlang.org/docs/reference/</a></li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	<p>Bachelor: Medieninformatik (Grundlagen, 2. Sem.)</p> <p>Bachelor: Informatik (Grundlagen, 2. Sem.)</p> <p>Bachelor: IT-Management (Grundlagen, 2. Sem.)</p> <p>Bachelor: Wirtschaftsinformatik (Grundlagen, 2. Sem.)</p>
Besonderheiten:	Die Veranstaltung ist als "Flex Classroom" konzipiert. Studierende können die Learning Outcomes durch Besuch der Vorlesung oder durch die Nutzung der bereitgestellten Screencasts erreichen. Die Praktika bestehen aus Beratung und Abnahme von Praktikumsaufgaben.
Letzte Aktualisierung:	20.10.2020

## 6.7 [Betriebssysteme und verteilte Systeme]

Modulnummer:	BS
Modulbezeichnung:	Betriebssysteme und verteilte Systeme
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	1

Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Lutz Köhler
Dozierende:	Prof. Dr. Lutz Köhler
Learning Outcome:	(WAS) Die Studierenden sollen die Prinzipien und Mechanismen von Betriebssystemen und verteilten Systemen verstanden haben, (WOMIT) indem am Beispiel von UNIX selbstständig Systemprogramme geschrieben und Betriebssystemstrukturen bewertet werden, (WOZU) um die Mechanismen zur Implementierung verteilter Anwendungen anwenden können.
Modulinhalte:	<ul style="list-style-type: none"> <li>• Das Betriebssystem UNIX</li> <li>• UNIX-Prozesse und elementare Kommunikation</li> <li>• Prozessmodell, Nebenläufigkeit und Synchronisation</li> <li>• Anwendungsszenarien zur Interprozess-Kommunikation</li> <li>• UNIX-Dateisystem</li> <li>• Verteilte Systeme</li> <li>• Strategien zum Scheduling und zur Speicherverwaltung</li> <li>• Systemprogrammierung am Beispiel von UNIX: <ul style="list-style-type: none"> <li>• Shell-Programmierung, Prozess-Modelle, Prozess-Erzeugung und Synchronisation,</li> <li>• UNIX-Prozesse und elementare Synchronisation, Pipes, Shared Memory,</li> <li>• Synchronisationsprimitive für den wechselseitigen Ausschluss, Semaphore,</li> <li>• Nachrichtenwarteschlangen,</li> <li>• Dateisysteme,</li> <li>• TCP/IP, Sockets, Remote Procedure Cal</li> <li>• Strategien zum Scheduling und zur Speicherverwaltung</li> <li>• Klassische Synchronisationsprobleme.</li> </ul> </li> </ul>
Lehr- und Lernmethoden:	4 SWS: Vorlesung 2 SWS; Praktikum 2 SWS
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150 h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	Keine über die Zulassungsvoraussetzungen hinausgehenden Voraussetzungen.
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	Systemprogrammierung in UNIX / Linux [Elektronische Ressource]: Grundlegende Betriebssystemkonzepte und praxisorientierte Anwendungen / von Erich Ehses, Lutz Köhler, Petra Riemer, Horst Stenzel, Frank Victor, Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2012 Moderne Betriebssysteme / Andrew S. Tanenbaum, Herbert Bos, 4. aktualisierte Auflage. - Hallbergmoos: Pearson, [2016]
Verwendung des Moduls in weiteren Studiengängen:	-
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.8 [BWL - Grundlagen]

Modulnummer: **BWL**

Modulbezeichnung:	BWL - Grundlagen
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	2
Häufigkeit des Angebots:	Jedes Semester
Modulverantwortliche*r:	Prof. Dr. Monika Engelen (Sommersemester), Prof. Dr. Torsten Klein (Wintersemester)
Dozierende:	Prof. Dr. Monika Engelen, Prof. Dr. Torsten Klein
Learning Outcome:	<p>(WAS) Die Studierenden kennen und verstehen die wichtigsten Entscheidungsbereiche wirtschaftlichen Handelns,</p> <p>(WOMIT) indem Sie</p> <ul style="list-style-type: none"> <li>• grundlegende Entscheidungen im Rahmen einer Unternehmensgründung beschreiben,</li> <li>• Aufgaben der Unternehmensführung, wie die Konzeption einer tragfähigen Strategie, kennen,</li> <li>• Aufgaben der Teilbereiche Produktion, Absatz und Marketing sowie Investition und Finanzierung verstehen,</li> <li>• Investitionsentscheidungen informationsgestützt treffen, sowie</li> <li>• Kalkulationsverfahren der Investitionsrechnung anwenden und auswerten,</li> </ul> <p>(WOZU) um für weitere BWL-Veranstaltungen Ihres Studiums vorbereitet zu sein und in ihrem Berufsleben wirtschaftliche Konzepte im Unternehmenskontext anzuwenden.</p>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Grundlagen</li> <li>• Unternehmensführung: Ziele, Planung und Entscheidung, Ausführung und Kontrolle</li> <li>• Investition und Finanzierung</li> <li>• Konstitutive Entscheidungen</li> <li>• Produktion</li> <li>• Absatz und Marketing</li> </ul>
Lehr- und Lernmethoden:	Lehrvortrag, seminaristische Vorlesung, Übung (gestützt durch live-Voting) und digitale Selbstlernkontrollen 4 SWS: Vorlesung: 2 SWS + Übung: 2
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	60h
Selbststudium:	90h
Empfohlene Voraussetzungen:	Keine über die Zulassungsvoraussetzungen hinausgehenden Voraussetzungen.
Zwingende Voraussetzungen:	keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Wöhe (2016): Einführung in die Allgemeine Betriebswirtschaftslehre, 26. Auflage (Auszüge)</li> <li>• Übungsunterlagen und weitere Materialien auf ILIAS im Kurs "Grundlagen der BWL (BWL 1)" im Grundstudium Ingenieurwissenschaften</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	<p>Bachelor: IT-Management (Grundlagen, 1. Sem.)</p> <p>Bachelor: Informatik (Grundlagen, 1. Sem.)</p> <p>Bachelor: Medieninformatik (Vertiefung, 3. Sem.)</p> <p>Bachelor: Ingenieurwissenschaftliches Grundstudium (Allgemeiner Maschinenbau, Elektrotechnik, Wirtschaftsingenieurwesen) (1. Sem)</p>
Besonderheiten:	keine



Letzte Aktualisierung:	20.10.2020
------------------------	------------

## 6.9 [Mathematik II]

Modulnummer:	MA2
Modulbezeichnung:	Mathematik II
Art des Moduls:	Pflicht
ECTS credits:	8
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	2
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Wolfgang Konen
Dozierende:	Prof. Dr. Wolfgang Konen
Learning Outcome:	<p>(WAS) Die Studierenden erwerben die Fähigkeiten zur Analyse realer oder geplanter Systeme,</p> <p>(WOMIT) indem sie praktische Aufgabenstellungen aus dem Informatik-Umfeld in mathematische Strukturen abstrahieren und lernen, selbstständig die Modellfindung und die Ergebnisbeurteilung vorzunehmen.</p> <p>(WOZU) Dabei sollen die Anwendungsbezüge der Mathematik deutlich werden, z.B. die Beziehungen diskreter Strukturen wie der Graphen zu vielfältigen grundlegenden Datenstrukturen, die Statistik zur Deskription und Beurteilung von Beobachtungen und die Analysis zur Verarbeitung von Signalen und zur Lösung von mathematischen Modellen.</p>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Analysis (mehrerer Veränderlichen)</li> <li>• Graphentheorie</li> <li>• Statistik und Wahrscheinlichkeitsrechnung</li> <li>• Komplexe Zahlen und</li> <li>• Differentialgleichungen</li> </ul>
Lehr- und Lernmethoden:	<ul style="list-style-type: none"> <li>• Vorlesung mit Skript und Tabletmitschrieb zum Download</li> <li>• Übungen mit vorbereitenden Aufgaben, die von Studenten vorgestellt und gemeinsam diskutiert werden, zur Vertiefung der Inhalte der Vorlesung</li> <li>• Projekt-Praktikum mit Teamarbeit, Dokumentation und Präsentation, sowie Abnahme Probeklausur-Aufgaben</li> <li>• fallweise Tutorien mit studentischen Tutor*innen zur Klärung Verständnisfragen</li> </ul>
Prüfungsformen:	Klausur
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	240h
Präsenzzeit:	110 h (4 V + 2 Ü + 1 P SWS)
Selbststudium:	130h
Empfohlene Voraussetzungen:	Keine über die Zulassungsvoraussetzungen zum Studium hinausgehenden. Der vorherige Besuch von Mathematik I ist sinnvoll, aber keine zwingende Voraussetzung.
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<p>Skript unter <a href="http://www.gm.fh-koeln.de/~konen/Mathe2-SS">www.gm.fh-koeln.de/~konen/Mathe2-SS</a>, dort auch weitere Literatur, z. B.:</p> <ul style="list-style-type: none"> <li>• <b>Teschl, Gerald und Teschl</b>, Susanne: "Mathematik für Informatiker", 4. Auflage, Springer Verlag, 2013</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Hartmann, Peter:</b> "Mathematik für Informatiker – Ein praxisbezogenes Lehrbuch", 7. Auflage, Vieweg Verlag, 2020</li> <li>• <b>Papula, Lothar:</b> "Mathematik für Ingenieure und Naturwissenschaftler" Vieweg Verlag, 2012</li> <li>• <b>Knorrenschild, Michael,</b> "Vorkurs Mathematik", 4. Auflage, Hanser-Verlag, 2013</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	<ul style="list-style-type: none"> <li>• Bachelor: IT-Management (Grundlagen, 2. Sem.)</li> <li>• Bachelor: Informatik (Grundlagen, 2. Sem.)</li> <li>• Bachelor: Medieninformatik (Grundlagen, 2. Sem.)</li> </ul>
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.10 [Theoretische Informatik II]

Modulnummer:	T12
Modulbezeichnung:	Theoretische Informatik II
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	Sommersemester
Häufigkeit des Angebots:	Jedes Semester
Modulverantwortliche*r:	Prof. Dr. Martin Eisemann
Dozierende:	Prof. Dr. Martin Eisemann
Learning Outcome:	<p>(WAS) Grundsätzliches Ziel des Kurses ist eine Einführung in die Begriffe, Methoden, Modelle und Arbeitsweise der Theoretischen Informatik anhand der ausgewählten Teilgebiete. Dabei lernen die Studierenden Probleme und Sachverhalte zu abstrahieren und zu modellieren (etwa logische und algebraische Kalküle, graphentheoretische Notationen, formale Sprachen und Automaten sowie spezielle Kalküle wie Petri-Netze).</p> <p>(WOMIT) Die Studierenden erwerben fundierte Kenntnisse der grundlegenden Themengebiete und eine wesentliche Basis und Vorbereitung für Veranstaltungen in höheren Semestern des Studiums. Aufgaben zu den Lehrinhalten (s.u.) werden in kleinen Gruppen (Teamarbeit) selbständig gelöst. Die Lösungen sollen in den Übungsstunden vorgetragen und der Lösungsweg den Kommilitonen hierbei erläutert werden.</p> <p>(WOZU) In verschiedenen Grundlagengebieten der Informatik lernen die Studierenden Verfahrensweisen kennen, um den algorithmischen Kern eines Problems zu identifizieren und können passende Algorithmen entwerfen (Automaten, Turing Maschinen, Logik). Dabei können Sie bekannte Problemstellungen im Anwendungskontext erkennen und sind mit den zugehörigen Lösungsmustern vertraut (Modellierung mittels Automaten, Petri-Netzen, Boolescher Algebra, etc.).</p>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Reguläre (Typ-3) Sprachen: Endliche Automaten, Reguläre Ausdrücke; Typ3-Grammatiken, Zustandsübergangsdiagramme; Chomsky-Hierarchie</li> <li>• Modellierung sequentieller und paralleler (Ausgabe-) Prozesse: Endliche Maschinen / Automaten; Automatenetze, Petri-Netze, Zelluläre Automaten</li> <li>• Kontextfreie (Typ-2) Sprachen: Kontextfreie Grammatiken, Chomsky-Normalform; Kellerautomaten; Anwendungen (Ableitungs- und Syntaxbäume, Syntax von Programmiersprachen, Backus-Naur-Form)</li> <li>• Kontextsensitive (Typ-1) und rekursiv aufzählende (Typ-0) Sprachen: Grammatiken, Turingautomaten, Einführung in die Begriffe: Berechenbarkeit, Entscheidbarkeit und Komplexität</li> </ul>
Lehr- und Lernmethoden:	Vorlesung mit Übung 4 SWS: Vorlesung 2 SWS; Übung 2 SWS

Prüfungsformen:	Klausur
Workload (25 - 30 h $\triangleq$ 1 ECTS credit) :	150h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	Keine über die Zulassungsvoraussetzungen hinausgehenden Voraussetzungen.
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Hoffmann, D. (2011): Theoretische Informatik, 2. Auflage</li> <li>• Vossen, G., Witt K. (2004): Grundlagen der Theoretischen Informatik mit Anwendungen. 3. Aufl. Vieweg &amp; Sohn, Braunschweig.</li> <li>• Hedtstück, U. ( 2004 ): Einführung in die Theoretische Informatik. Oldenbourg, München.</li> <li>• Asteroth, A., Baier, C. (2002) Theoretische Informatik. Pearson Studium München</li> <li>• Hopcroft, J. E. et al. (2002): Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie. Pearson Studium, München.</li> <li>• Schöning, U. (1997): Theoretische Informatik - kurzgefaßt. 3. Aufl. Spektrum Akademischer Verlag, Heidelberg.</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Bachelor: Medieninformatik (Grundlagen, 2. Sem.)
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.11 [Algorithmik]

Modulnummer:	ALG
Modulbezeichnung:	Algorithmik
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	3
Häufigkeit des Angebots:	Jedes Wintersemester
Modulverantwortliche*r:	Prof. Dr. Heinrich Klocke
Dozierende:	Prof. Dr. Heinrich Klocke
Learning Outcome:	<p>(WAS) Die Studierenden sollen in der Lage sein, die Zeitkomplexität von Algorithmen asymptotisch zu analysieren, experimentelle Laufzeitanalysen mit verschiedenen Testmethoden wie Ratiotest, Powertest, durchzuführen, auszuwerten und zu dokumentieren.</p> <p>Die Kern- und Laufzeiteigenschaften sowie die Unterschiede verschiedener Algorithmenklassen wie Divide&amp;Conquer, Greedy, Backtracking, dyn. Programmierung sollen tief verstanden und anhand typischer algorithmischer Beispiele erklärt werden können.</p> <p>Die Studierenden müssen das Prinzip der Induktion im Zusammenhang mit der Konstruktion rekursiver Algorithmen verstehen und praktisch bei konkreten Aufgabenstellungen anwenden können.</p> <p>(WOMIT) Ein fundiertes theoretisches Verständnis für den Zusammenhang zwischen Datenstruktur, Algorithmus und asymptotischer Laufzeit soll aufgebaut werden.</p>

	<p>Die Studierenden sollen sich durch die Vorlesung und eigenes weiterführendes Literaturstudium das grundlegende algorithmische Wissen erwerben, um in den Übungen und Praktika konkrete praxisnahe algorithmische Aufgaben in Teams lösen zu können.</p> <p>(WOZU) Besonders wichtig ist die Fähigkeit zu erkennen, wann und wie bekannte und wohl untersuchte Datenstrukturen und Algorithmen durch geeignete Modifikationen und/oder Erweiterungen auf praktische algorithmische Probleme und Aufgaben angewendet werden können.</p>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Asymptotische Analyse, O-Notation, Mastertheorem</li> <li>• ADT's u. algebraische Spezifikation für Mengen, Tabellen und Dictionaries</li> <li>• Dictionaries</li> <li>• Binäre Suchbäume</li> <li>• Balancierte Bäume</li> <li>• AVL-Bäume</li> <li>• Rot-Schwarz-Bäume</li> <li>• kd-Bäume</li> <li>• Bayer-Bäume</li> <li>• Hash-Techniken</li> <li>• Hashfunktionen</li> <li>• Sondierungstechniken</li> <li>• Universelles Hashing</li> <li>• Priority Queues</li> <li>• Binäre Heaps, Heapsort</li> <li>• Binomialheaps</li> <li>• Fibonacciheaps</li> <li>• Divide&amp;Conquer-Algorithmen</li> <li>• MergeSort, Randomized Quickselect, etc.</li> <li>• Differenzgleichungen</li> <li>• Mastertheorem</li> <li>• Greedy-Algorithmen</li> <li>• Graph-Algorithmen</li> <li>• Induktionsprinzip am Beispiel Eulerscher Graphen</li> <li>• Basisalgorithmen: Tiefen- u. Breitensuche</li> <li>• Topologisches Sortieren</li> <li>• Kürzeste-Wege-Algorithmen (Dijkstra, Floyd-Warshall, etc.)</li> <li>• Transitive Hülle von Graphen</li> <li>• Spannende Bäume mit minimalen Kosten (Kruskal, Prim)</li> <li>• Flüsse in Netzwerken (Ford u. Fulkerson)</li> <li>• Dynamische Routing-Algorithmen in Netzen (Bellman-Ford)</li> <li>• Dynamische Programmierung (DP)</li> <li>• Rucksack-Problem</li> <li>• Matrixketten-Multiplikation</li> <li>• Scheduling-Algorithmen</li> <li>• Sequence Alignment</li> <li>• Typische Konstruktionsschritte bei der DP</li> <li>• Vergleichende Diskussion von DP- (bottom-up) und Greedy-Algorithmen (top-down)</li> <li>• Informierte Suchstrategien (A*)</li> <li>• Randomisierte Algorithmen (optional)</li> </ul>
Lehr- und Lernmethoden:	<p>Die Einführung und Erläuterung der Eigenschaften, der Anwendungsmöglichkeiten von Algorithmen und Datenstrukturen sowie von Pseudocode findet anhand von Folien statt.</p> <p>Der Ablauf von Algorithmen wird auf einem digitalen Medium wie z.B. iPad Schritt für Schritt handschriftlich anhand konkreter Beispiele ausgeführt. Die Studierenden können so die einzelnen algorithmischen Abläufe in kleinen Schritten und mit genügend Zeit verfolgen.</p> <p>4 SWS: Vorlesung: 2 SWS + Praktikum: 1 SWS + Übung: 1 SWS / 15h</p>
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	Modul Algorithmen und Programmierung I u. II erfolgreich abgeschlossen. Mathematik erfolgreich abgeschlossen. Programmieren können.
Zwingende Voraussetzungen:	Keine

Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Thomas Cormen, Charles E. Leiserson, Ronald E. Rivest, Clifford Stein. Algorithmen - Eine Einführung. De Gruyter Oldenbourg. 2013 4. Auflage ISBN-13: 978-3486748611</li> <li>• Sara Baase, Allen Van Gelder. Computer Algorithms. 3rd Edition Addison Wesley 2000, ISBN: 0201612445</li> <li>• Michael T. Goodrich, Roberto Tamassia. Algorithm Design. Wiley 2002. ISBN 0471383651</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	-
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.12 [Datenbanksysteme]

Modulnummer:	DBS
Modulbezeichnung:	Datenbanksysteme
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	3
Häufigkeit des Angebots:	Jedes Wintersemester
Modulverantwortliche*r:	Prof. Dr. Birgit Bertelsmeier
Dozierende:	Prof. Dr. Heide Faeskorn-Woyke, Prof. Dr. Birgit Bertelsmeier
Learning Outcome:	(WAS) Die Studierenden sollen über ein einheitliches konsistentes Begriffsgebäude bezüglich der Datenbankthematik verfügen, (Womit) indem sie die theoretischen Grundlagen von Datenbanksystemen am Beispiel relationaler und objektrelationaler Datenbanksysteme verstanden haben, insbesondere die relationale Algebra, die Normalisierung sowie funktionale Abhängigkeiten und in der Lage sind, diese Erkenntnisse im Rahmen der Modellierung, Normalisierung und Implementierung von Datenbankschemata praktisch anzuwenden, (Wozu) um komplexere Datenbankanfragen, Datendefinitionen und Datenänderungen über SQL programmieren können, mit dem Transaktionsbegriff, der Mehrbenutzersynchronisation und Verfahren zur Fehlererholung sowie zur Sicherung und der Datenintegrität vertraut sind und Aufgaben der Integritätsprüfung praktisch lösen können und ein erstes Verständnis für SQL-Tuning in relationalen Datenbanksystemen entwickeln.
Modulinhalte:	<ul style="list-style-type: none"> <li>• Grundbegriffe und Architektur von Datenbanken</li> <li>• Ein Vorgehensmodell zur Erstellung eines Datenbanksystems</li> <li>• Datenmodellierung (Entity Relationship Modell) und Implementierung am Beispiel eines relationalen Datenbanksystems</li> <li>• Grundlagen des relationalen Modells</li> <li>• Funktionale Abhängigkeiten und Normalisierung</li> <li>• Relationale Algebra und Anfrageoptimierung</li> <li>• Datenintegrität</li> <li>• Datenbanksprache SQL: DDL, DML, DAL, Integritätsbedingungen und Constraints unter dem jeweils aktuellen SQLStandard, zur Zeit SQL2016</li> <li>• Transaktionskonzepte, Mehrbenutzersynchronisation, Fehlererholung und Datensicherheit</li> <li>• Tuning von SQL-Anfragen und Schemadesign</li> </ul>
Lehr- und Lernmethoden:	Folien mit Erklärungen, Syntax-Diagrammen und vielen Beispielen zur Wissensvermittlung und als Lernbasis für zu Hause interaktive Erarbeitung von Vorgehensweisen und Lösungen für Aufgaben während der Vorlesung und Übung, das Tutorium hilft bei individuellen Fragen weiter eLearning-Datenbank-Portal der TH Köln mit verschiedenen interaktiven Trainern und Multiple-Choice-Tests ( <a href="https://edb2.th-koeln.de/">https://edb2.th-koeln.de/</a> )

	Lernvideos von der Übung (sebstaufgezeichnet Herr Damian Gawenda), Lernvideos von Prof. Dr., Jens Dittrich, Universität des Saarlandes, Fakultät für Mathematik und Informatik: <a href="https://www.youtube.com/user/jensdit">https://www.youtube.com/user/jensdit</a> und <a href="https://bigdata.uni-saarland.de/datenbankenlernen">https://bigdata.uni-saarland.de/datenbankenlernen</a> Video-Aufzeichnung der Vorlesung 5 SWS (Vorlesung: 2 SWS , Praktikum: 1 SWS , Übung: 1 SWS , Seminar: 1 SWS )
Prüfungsformen:	Erfolgreiche Teilnahme am Praktikum als Voraussetzung für die Zulassung zur Klausur  Klausur
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	150h
Präsenzzeit:	30 h Vorlesung, 5h Praktikum, 15h Übung
Selbststudium:	100h
Empfohlene Voraussetzungen:	Wissen und Fähigkeiten, wie sie in den Modulen Einführung in die Informatik, Mathematik I + II, Algorithmen und Programmierung I + II vermittelt werden.
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Brücher, C., Jüdes, F., Kollmann, W.: "Oracle SQL Thinking –VomProblem zum-SQL-Statement mitOracle 12c", 2014, MITP, 978-3-8266-9581-0, (e-Book)</li> <li>• Elmasri R., Navathe, S. B. : "Fundamentals of Database Systems, Global Edition", Media-Kombination, 1272 Seiten, 2016, 7th edition, Pearson Education Limited, 978-1-292-09761-9</li> <li>• Faeskorn-Woyke, H., Bertelsmeier, B., Riemer, P., Bauer, E.: „Datenbanksysteme: Theorie und Praxis mit Oracle und MySQL“, Pearson, 2011 - als pdf zum Download bereitgestellt</li> <li>• Kemper, A., Eickler, A.: "Datenbanksysteme – Eine Einführung", 2015, 10. Auflage, De Gruyter, 978-3-11-044375-2</li> <li>• Laube, M.: "Einstieg in SQL - Für alle wichtigen Datenbanksysteme: MySQL, PostgreSQL, MariaDB, MS SQL. Ohne Vorkenntnisse einsteigen!", 2019, 2. Auflage, Rheinwerk, 978-3-8362-7070-0</li> <li>• Saake, K.-U. Sattler, A. Heuer: "Datenbanken – Konzepte und Sprachen", 2018, 6. Auflage, MITP, 978-3-95845-776-8, ((e)-Book)</li> <li>• Schicker, E.: "Datenbanken und SQL - Eine praxisorientierte Einführung mit Anwendungen in Oracle, SQL Server und MySQL", 2017, 5. Auflage, Springer Vieweg, 978-3-658-16128-6</li> <li>• Studer, Th.: "Relationale Datenbanken - Von den theoretischen Grundlagen zu Anwendungen mit PostgreSQL", 2019, 2. Auflage, 2019, Springer Berlin, 978-3-662-58975-5</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Bachelor: Wirtschaftsinformatik (Vertiefung, 3. Sem.) Bachelor: Medieninformatik (Vertiefung, 3. Sem.) Bachelor: IT-Management (Vertiefung, 3. Sem.)
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

### 6.13 [Informations- und Prozessmanagement]

Modulnummer:	IMP
Modulbezeichnung:	Informations- und Prozessmanagement
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	3
Häufigkeit des Angebots:	Jedes Wintersemester

Modulverantwortliche*r:	Prof. Dr. Matthias Zapp
Dozierende:	Prof. Dr. Matthias Zapp
Learning Outcome:	<p>(Was) Studierenden sollen befähigt werden betriebliche Abläufe in Unternehmen zu analysieren und mit Hilfe von Informationstechnologie zu optimieren,</p> <p>(Womit) indem sie</p> <ul style="list-style-type: none"> <li>• moderne Konzepte des Informationsmanagements und hieraus resultierende Anforderungen an Informationssystemen im betrieblichen Umfeld verstehen</li> <li>• heterogene Geschäftsprozesse und deren potentielle Unterstützung mit Informationstechnologie in Zusammenarbeit mit Fachvertretern analysieren</li> <li>• Geschäftsprozesse mit Hilfe von Modellierungssprachen präzise abbilden und zusammen mit Fachvertretern optimieren</li> <li>• Werkzeuge der (Teil-) Prozesseautomatisierung anwenden,</li> </ul> <p>(Wozu) um später Geschäftsprozesse in unterschiedlichen Anwendungsdomänen durch Informationstechnologie zu optimieren und automatisieren.</p>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Konzepte des Informationsmanagement in Unternehmen</li> <li>• Herausforderungen und Trends für ein effektives Informationsmanagement innerhalb von Unternehmen und Wertschöpfungsketten</li> <li>• Moderne Konzepte des Geschäftsprozessmanagement / Business Engineering</li> <li>• Modellierungssprachen für Geschäftsprozesse wie BPMN2</li> <li>• Werkzeuge zur Prozessmodellierung und automatisierung</li> </ul>
Lehr- und Lernmethoden:	Vorlesung und fallstudienbasierte Übung (2 SWS Vorlesung, 1 SWS Übung) Einsatz von Screencasts und Tools zum Live-Voting und selbständiger Lernkontrolle
Prüfungsformen:	Klausur
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	150 h
Präsenzzeit:	45h
Selbststudium:	105h
Empfohlene Voraussetzungen:	keine über die Zulassungsbedingungen hinausgehenden Voraussetzungen
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<p>Basisliteratur Geschäftsprozessmanagement:</p> <ul style="list-style-type: none"> <li>• <b>Allweyer, T.:</b> Geschäftsprozessmanagement: Strategie, Entwurf, Implementierung, Controlling. Herdecke.</li> <li>• <b>Allweyer T.:</b> BPMN 2.0 - Business Process Model and Notation. Einführung in den Standard zur Geschäftsprozessmodellierung. Norderstedt.</li> </ul> <p>Fokus Geschäftsprozessautomatisierung</p> <ul style="list-style-type: none"> <li>• <b>Freund, T.; Rücker, B.:</b> Praxishandbuch BPMN Geschäftsprozessmanagement: Mit Einführung in CMMN und DMN. München.</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Bachelor: Wirtschaftsinformatik Bachelor: Informatik
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.14 [Kommunikationstechnik und Netze]

Modulnummer:	KTN
Modulbezeichnung:	Kommunikationstechnik und Netze
Art des Moduls:	Pflicht
ECTS credits:	5

Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	3
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Hans Ludwig Stahl
Dozierende:	Prof. Dr. Hans Ludwig Stahl
Learning Outcome:	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>• (Was und Womit) eigenen sich ein breites Spektrum von einschlägigem Grundlagenwissen und wichtigen Grundbegriffen an, (Wozu) das sie befähigt, Aufgaben- und Problemstellungen, Analyseverfahren und -ergebnisse sowie Lösungen qualifiziert zu beschreiben, zu spezifizieren und auf Expertenniveau zu diskutieren und zu dokumentieren</li> <li>• (Was) verstehen wichtige Kommunikationsmodelle und die Prinzipien protokollbasierter Kommunikation, (Womit) indem sie die Konzepte der in der Lehrveranstaltung vorgestellten Protokollwelten auf geeignete Problemstellungen anwenden, (Wozu) um spezifische Kommunikationsaufgaben zu lösen</li> <li>• (Was) lernen Verfahren und Werkzeuge der Netz- und Protokollanalyse kennen, (Womit) indem sie die in der Vorlesung und im Praktikum behandelten Verfahren und Werkzeuge geeignet auswählen und konfigurieren können, (Wozu) um diese auf angemessene Weise zur Analyse bestimmter Problemstellungen einzusetzen und damit zielführend zu Problemlösungen beizutragen</li> <li>• (Was) beherrschen die in der Vorlesung vorgestellte grundlegende Terminologie der IT-Sicherheit, (Womit) indem sie typische Sicherheitsmaßnahmen für Rechnernetze kennen und verstehen, (Wozu) wodurch sie in der Lage sind, vorgestellte Beispielszenarien zu analysieren und mögliche Gegenmaßnahmen zum Schutz vor erkannten Risiken ableiten zu können</li> <li>• (Was und Womit) kennen die wesentlichen Aspekte und Eigenschaften der Internet-Protokolle IPv4 und IPv6 sowie der damit verbundenen Protokollwelten und (Wozu) sind in der Lage, auf Basis einer qualifizierten Anforderungsanalyse diese auf geeignete Weise in die Realisierung von Kommunikations- und Vernetzungsaufgaben einzubeziehen</li> </ul>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Grundbegriffe und Grundlagen: Kommunikationssysteme (Modelle, Grundbegriffe); Protokolle, Schnittstellen, Dienste; Architekturmodelle (OSI-Referenzmodell, TCP/IP-Protokollfamilie); Standardisierung (ISO, ANSI, DIN, IETF, ...)</li> <li>• Die TCP/IP-Protokollfamilie als Grundlage des Internet: Wichtigste Protokolle der TCP/IP-Protokollfamilie; Schichtenmodell und Protokolle im Detail; Adressierung auf den verschiedenen Ebenen; ausgewählte Anwendungen; Klassifizierung von Netzen, Topologien, Technologien</li> <li>• Wegewahl / Vermittlung / Routing: Wegewahl und Routing; Vermittlungsprinzipien; Routing-Verfahren und Protokolle; Internetspezifische Verfahren</li> <li>• Einführung in die Netzsicherheit: grundlegende Begriffe der IT-Sicherheit; typische Bedrohungen in IP-basierten Netzen</li> <li>• Next Generation Internet: IPv6, weitere neue Entwicklungen; Neues in IPv6 im Vergleich zu IPv4, Unterschiede zwischen IPv4 und IPv6</li> <li>• Praktische Aspekte: Erlernen und Anwenden einfacher Netzanalyse und kontrollwerkzeuge; Untersuchung von typischen Protokollmechanismen</li> </ul>
Lehr- und Lernmethoden:	<p>Vorlesung, Praktikum an Rechnern des KTDS-Labors (Ressourcen: Netzanalysesoftware, div. Netzüberwachungssoftware, E-Mail- Server und Clients, DNS-Server, ggf. weitere Server-Implementierungen)</p> <p>4 SWS: Vorlesung: 3 SWS + Praktikum: 1 SWS</p>
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	72h (Vorlesung, Praktikum)
Selbststudium:	78h
Empfohlene Voraussetzungen:	Einschlägige Kenntnisse aus den vorhergehenden Semestern.
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Vorlesungsunterlagen: Foliensammlung, Beispiellösungen</li> </ul>



	<ul style="list-style-type: none"> <li>• Quellen im WWW: RFCs, Informationen zu den behandelten Protokollen und zu Implementierungsaspekten</li> <li>• Fachliteratur u. a.</li> <li>• Douglas E. Comer: „Computernetzwerke und Internets“</li> <li>• James F. Kurose, Keith W. Ross: „Computernetze“</li> <li>• Larry L. Peterson, Bruce S. Davie: „Computernetze“</li> <li>• Stephan Rupp, Gerd Siegmund, Wolfgang Lautenschläger: „SIP – multimediale Dienste im Internet“</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Bachelor: Wirtschaftsinformatik (Vertiefung, 3. Sem.) Bachelor: Medieninformatik (Vertiefung, 3. Sem.) Bachelor: IT-Management (Vertiefung, 3. Sem.)
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.15 [Projektmanagement]

Modulnummer:	PM
Modulbezeichnung:	Projektmanagement
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	3
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Lutz Köhler
Dozierende:	Prof. Dr. Holger Günther, Prof. Dr. Lutz Köhler, Prof. Dr. Mario Winter
Learning Outcome:	<p>Die Studierenden sollen befähigt werden,</p> <ul style="list-style-type: none"> <li>• (Was) die grundlegenden Aufgaben des Projektmanagements, insb. in IT-Projekten, zu charakterisieren und durchzuführen,</li> <li>• (Womit) indem sie Projektmanagement-Methoden, -Techniken und -Werkzeuge zur Organisation, Planung und Steuerung herkömmlicher und agiler Projekte zielgerichtet einzusetzen und</li> <li>• die erforderlichen soziologischen und kommunikativen Aspekte zu berücksichtigen,</li> </ul> <p>(Wozu) um insb. mit dem Ziel einer menschengerechten und soziologisch fundierten Menschenführung zur Erreichung einer wirklichen und optimalen Produktivität bei komplexen Projekten beitragen zu können.</p>
Modulinhalte:	<p>Das Modul befasst sich mit den Managementaspekten der professionellen Entwicklung großer Softwaresysteme.</p> <p>Der Vorlesungsteil des Moduls gliedert sich in folgende Kapitel:</p> <ul style="list-style-type: none"> <li>• Überblick – Warum Projektmanagement?</li> <li>• Teamarbeit und Menschenführung (Kommunikation und Führung)</li> <li>• Kosten/Nutzen-Analysen und Entscheidungstechniken</li> <li>• Projektorganisation und Projektplanung (Aufbauorganisation, Ablauforganisation, Prozessmodellierung, Netzplantechnik)</li> <li>• Detaillierte Aufwandsschätzung und Projektcontrolling (Function Point Analysis, COCOMO, Risikomanagement, Projektpräsentationen);</li> <li>• Inhalte PM-BOK (Project Management - Body of Knowledge);</li> <li>• Zusammenfassung und Prüfungsvorbereitung;</li> </ul> <p>Damit die Studierenden die vorgestellten Methoden und Techniken zum Management von Softwareprojekten anwenden sowie besser analysieren und bewerten können, werden im Praktikum die in der Vorlesung vermittelten Inhalte in Teams anhand von in Absprache</p>

	mit den Dozenten selbst gewählten Fallbeispielen eingesetzt. Dazu bilden die Teilnehmenden Teams zu jeweils 6 Studierenden. Im Praktikum werden folgende Bereiche vertieft:
	<ul style="list-style-type: none"> <li>• Kosten- Nutzenrechnung, Entscheidungstechniken;</li> <li>• Aufbauorganisation;</li> <li>• Aufwandsschätzung (Function-Point-Analyse, COCOMO);</li> <li>• Risikomanagement;</li> <li>• Ablauf- und Ressourcenplanung (Netzplantechnik, Einsatz von PM-Software wie z.B. MS-Project).</li> </ul>
Lehr- und Lernmethoden:	Beamer-gestützte Vorlesungen (Folien in elektronischer Form im Netz); Vertiefende Unterlagen sowie aktuelle Artikel (in elektronischer Form im Netz); Projektarbeit und Präsentationen in Kleingruppen, um die erlernten Methoden und Techniken einzuüben und zu vertiefen; 4 SWS: Vorlesung: 2 SWS + Praktikum: 1 SWS + Übung: 1 SWS
Prüfungsformen:	Projekt-Ausarbeitung; Vortrag; schriftlicher Test.
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	75h
Selbststudium:	75h
Empfohlene Voraussetzungen:	Keine
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• C. Aichele, M. Schönberger: IT-Projektmanagement. Springer Vieweg, 2014</li> <li>• Buhl: Grundkurs Projektmanagement. Carl Hanser Verlag, München, 2004</li> <li>• M. Broy, M. Kuhmann: Projektorganisation und Management im Software Engineering. Springer Vieweg, 2013</li> <li>• Hindel et Al.: Basiswissen Software-Projektmanagement. 2. Auflage, dpunkt-verlag, Heidelberg, 2006</li> <li>• H. Kerzner: Projektmanagement – Ein systemorientierter Ansatz. mitp-Verlag, Bonn, 2003</li> <li>• T. DeMarco: Spielräume - Projektmanagement jenseits von Burn-Out, Stress und Effizienz-Wahn. Hanser-Verlag, München, 2001</li> <li>• T. DeMarco: Der Termin - Ein Roman über Projektmanagement, Hanser-Verlag, München, Wien, 1998</li> <li>• T. DeMarco, T. Lister: Wien wartet auf Dich! (engl.: Peopleware); Hanser-Verlag, München, Wien, 1994</li> <li>• Project Management – Body of Knowledge. Project Management Institute, 1996</li> <li>• H.W. Wiczorrek, P. Mertens: Management von IT-Projekten Von der Planung zur Realisierung, 4. Aufl., Springer, Heidelberg, 2011</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Bachelor: Wirtschaftsinformatik (Vertiefung, 3. Sem.) Bachelor: Medieninformatik (Vertiefung, 5. Sem.) Bachelor: IT-Management (Vertiefung, 3. Sem.)
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.16 [Softwaretechnik I]

Modulnummer:	ST1
Modulbezeichnung:	Softwaretechnik I
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester

Empfohlenes Studiensemester:	3
Häufigkeit des Angebots:	Jedes Wintersemester
Modulverantwortliche*r:	Prof. Dr. Stefan Bente
Dozierende:	Prof. Dr. Stefan Bente
Learning Outcome:	<p>(Was) Als SW-Entwickler oder SW-Architekt die Anforderungen, das fachliche Domänenmodell und eine erste Grobarchitektur für ein Softwaresystem iterativ gestalten,</p> <p>(Womit) indem ich Anwendungsfälle (<b>Use Cases</b>) und Geschäftsobjekte im Fachlichen Datenmodell als <b>Klassendiagramm</b> aus der Aufgabenstellung ableite, die unterstützten Geschäftsprozesse des geplanten IT-Systems in Form von <b>Aktivitätsdiagrammen</b> spezifiziere, und die Transaktionen des geplanten Systems durch <b>Zustandsmodellierung</b> der Geschäftsobjekte erfasse, die Implementierung durch die Transformation des Fachlichen in das Logische Datenmodell vorbereite, und die Grobarchitektur des Systems in Form von <b>Komponentendiagrammen</b> spezifiziere, sowie die Komponenten-Interaktion als <b>Sequenz- und Kommunikations-Diagramme</b> abbilde, (Wozu) so dass die Grundlage für die weitere, mehr technisch orientierte Architektur- und Implementierungsarbeit gelegt ist.</p>
Modulinhalte:	<p>Einführung</p> <p>Modellierung fachlicher Anforderungen</p> <ul style="list-style-type: none"> <li>• Use Cases</li> <li>• Fachliches Datenmodell</li> </ul> <p>Verfeinerung der fachlichen Anforderungen: Verhaltensmodellierung</p> <ul style="list-style-type: none"> <li>• 4-Sichten-Modell</li> <li>• Geschäftsprozesse, Aktivitätsdiagramme</li> <li>• Geschäftsobjekt-Zustände, Zustandsdiagramme</li> </ul> <p>Vorbereitung Implementierung: Fachliches =&gt; Logisches Datenmodell</p> <ul style="list-style-type: none"> <li>• Logisches Datenmodell</li> <li>• Transition Fachliches =&gt; Logisches Datenmodell</li> </ul> <p>Entwurf der Grobarchitektur</p> <ul style="list-style-type: none"> <li>• 5 elementare Prinzipien</li> <li>• Clustering von Use Cases und Logischem Datenmodell</li> <li>• Komponentendiagramme, Komponenten-Schnittstellen</li> <li>• Komponenten-Interaktion: Sequenz- und Kommunikationsdiagramme</li> <li>• Architekturbewertung</li> </ul>
Lehr- und Lernmethoden:	Vorlesung, Praktikum, ggfs. mit Programmieranteil 4 SWS: 2 SWS Vorlesung + 2 SWS Praktikum
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	Algorithmen und Programmierung I und II , Theoretische Informatik
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Balzert, Helmut. <i>Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb</i>. 3. Aufl. 2012. Heidelberg: Spektrum Akademischer Verlag, 2011.</li> <li>• Booch, Grady, ed. <i>Object-Oriented Analysis and Design with Applications</i>. 3rd ed. The Addison-Wesley Object Technology Series. Upper Saddle River, NJ: Addison-Wesley, 2007.</li> <li>• Cockburn, A. (2000). <i>Writing Effective Use Cases</i>. Boston: Addison Wesley.</li> <li>• Hitz, Martin, Gerti Kappel, Elisabeth Kapsammer, and Werner Retschitzegger. <i>UML @ Work. Objektorientierte Modellierung mit UML 2. 3.</i>, akt. u. überarb. Aufl. Heidelberg: dpunkt, 2005.</li> <li>• Jochum, Friedbert: Vorlesungsscript Softwaretechnik 1 und 2. TH Köln, 2015 OMG. "OMG Unified Modeling Language, Version 2.5," 2015. <a href="http://www.omg.org/spec/UML/2.5/">http://www.omg.org/spec/UML/2.5/</a>.</li> <li>• Reussner, Ralf, and Wilhelm Hasselbring. <i>Handbuch der Software-Architektur</i>. 2., überarb. und erw. Auflage. Heidelberg: dpunkt Verlag, 2008.</li> <li>• Rupp, Chris, Stefan Queins, and die SOPHISTen. <i>UML 2 glasklar: Praxiswissen für die UML-Modellierung</i>. 4., aktualisierte und erweiterte Auflage. München: Carl Hanser Verlag GmbH &amp; Co. KG, 2012.</li> </ul>

	<ul style="list-style-type: none"> <li>• Sommerville, Ian. <i>Software Engineering</i>. 9. Aktual. München: Pearson Studium, 2012.</li> <li>• Toth, Stefan. <i>Vorgehensmuster für Softwarearchitektur: Kombinierbare Praktiken in Zeiten von Agile und Lean</i>. 2., aktualisierte und erweiterte Auflage. München: Carl Hanser Verlag GmbH &amp; Co. KG, 2015.</li> <li>• Zörner, Stefan. <i>Softwarearchitekturen dokumentieren und kommunizieren</i>. 2., überarbeitete und erweiterte Auflage. München: Carl Hanser Verlag GmbH &amp; Co. KG, 2015.</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	-
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.17 [Informatik, Recht und Gesellschaft]

Modulnummer:	IRG
Modulbezeichnung:	Informatik, Recht und Gesellschaft
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	4
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Mario Winter
Dozierende:	Prof. Dr. Mario Winter, Julia-Isabell Henke
Learning Outcome:	<p>(Was) Informatikerinnen und Informatiker analysieren und konstruieren sozio-technische Systeme und entwickeln dabei semiotische Artefakte wie z.B. Spezifikationen, Programme und Handbücher. Die entwickelten Systeme bilden einerseits soziale Wirklichkeit in vielfältiger Form ab und ändern andererseits diese Wirklichkeit durch ihren Einsatz.</p> <p>(Womit) Die Studierenden sollen befähigt werden,</p> <ul style="list-style-type: none"> <li>• die unterschiedlichen Wechselwirkungen zwischen Informatik-Systemen und ihrem Einsatzumfeld zu erkennen und zu bewerten,</li> <li>• ethische und datenschutz-rechtliche Aspekte des Einsatzes von Informatik-Systemen zu charakterisieren,</li> <li>• die Grundbegriffe des deutschen Privatrechts zu verstehen, und</li> <li>• sich im dazugehörigen Gesetzeswerk zu orientieren,</li> </ul> <p>(Wozu) um ein kritisches Bewusstsein für die aktuellen Fragen des wechselseitigen Einflusses von Informatik und Gesellschaft zu entwickeln und insbesondere im Bereich des Vertragsrechts selbständige Lösungsvorschläge erarbeiten zu können.</p>
Modulinhalte:	<p>Lehrveranstaltung Informatik und Gesellschaft (IUG, Ringveranstaltung geleitet von Prof. Dr. Mario Winter):</p> <ul style="list-style-type: none"> <li>• Die Wechselwirkungen zwischen den von Informatikern entwickelten Systemen und ihrem Einsatzumfeld werden in drei großen Themenblöcken behandelt:</li> <li>• Informatik und soziale Kontexte</li> <li>• Komplexität und Sicherheit in sozio-technischen Systemen</li> <li>• Systemgestaltung und Verantwortung der Informatik.</li> <li>• Beispielhafte Inhalte: Geschichte der Informatik, Bildung und Wissenschaft, Wissenschaften und Gesellschaft, Digitale Medien und Internet, Datenschutz und Überwachungstechniken, Informatik und Gestaltung, partizipative Systemgestaltung, Open Source, Ethische Leitlinien für Informatiker, Normen und Standards, philosophische Aspekte der Informatik.</li> </ul> <p>Lehrveranstaltung Recht (RE, Dozent: Julia-Isabell Henke):</p> <ul style="list-style-type: none"> <li>• Einführung in das deutsche Privatrecht, insbesondere in das BGB.</li> </ul>

	<ul style="list-style-type: none"> <li>• Schwerpunkt im Schuldrecht, hier insbesondere im Vertragsrecht.</li> <li>• Besondere Aspekte des Verbraucherschutzes und der inhaltlichen Gestaltung von Verträgen.</li> <li>• Im Allgemeinen Teil des BGB wird auf den Vertragsschluss, die Willenerklärung als rechtsgeschäftliches Gestaltungsmittel und die allgemeinen Anforderungen an die Vertragspartner eingegangen.</li> </ul>
Lehr- und Lernmethoden:	Beamer-gestützte Vorlesungen (Folien in elektronischer Form im Netz) Vertiefende Unterlagen sowie aktuelle Artikel (in elektronischer Form im Netz) Seminar in Gruppenarbeit, um ausgewählte Themen zu vertiefen und zu präsentieren
Prüfungsformen:	IUG: Klausur (90 Min) RE: Klausur (60 Min.)
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	Abgeschlossenes Grundstudium, Sonst keine besonderen Voraussetzungen
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<p>IUG:</p> <ul style="list-style-type: none"> <li>• Sara Baase: A Gift of Fire. Social, Legal, and Ethical Issues in Computing. Prentice Hall, Upper Saddle River, 1997</li> <li>• A.F. Chalmers: Wege der Wissenschaft. 5. Aufl., Springer, Heidelberg, 2001</li> <li>• D.M. Hester, P.J. Ford: Computers and Ethics in the Cyberage. Prentice Hall, Upper Saddle River, 2001</li> <li>• P. Gola, C. Klug: Grundzüge des Datenschutzrechts. C.H. Beck, 2003</li> <li>• M. Pierson, D. Seiler: Internet-Recht im Unternehmen. Beck-Rechtsberater im dtv, Deutscher Taschenbuch Verlag, München, 2002</li> <li>• <a href="http://www.gi-ev.de">http://www.gi-ev.de</a> Arbeitskreis Informatik und Verantwortung, Ethische Leitlinien der GI</li> <li>• <a href="http://www.bfd.bund.de">http://www.bfd.bund.de</a> Der Bundesbeauftragte für den Datenschutz</li> <li>• <a href="http://www.aktiv.org/DVD">http://www.aktiv.org/DVD</a> Deutsche Vereinigung für Datenschutz</li> <li>• <a href="http://www.big-brother-award.org">http://www.big-brother-award.org</a> Überwachungsinformationen</li> </ul> <p>Recht:</p> <ul style="list-style-type: none"> <li>• Bürgerliches Gesetzbuch in der aktuellen Taschenbuchausgabe des dtv</li> </ul> <p>Fakultativ:</p> <ul style="list-style-type: none"> <li>• Eugen Klunziger, Einführung in das Bürgerliche Recht, Verlag Vahlen</li> <li>• Norbert Ullrich, Wirtschaftsrecht für Betriebswirte, Verlag Neue Wirtschaftsbriefe</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Bachelor: Medieninformatik Bachelor: IT-Management Bachelor: Wirtschaftsinformatik
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.18 [Künstliche Intelligenz]

Modulnummer:	KI
Modulbezeichnung:	Künstliche Intelligenz
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	4

Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Heinrich Klocke
Dozierende:	Prof. Dr. Heinrich Klocke
Learning Outcome:	<ul style="list-style-type: none"> <li>• (Was) Die Studenten sollen die Methodik der Künstlichen Intelligenz und die Teilgebiete der KI verstehen lernen.</li> <li>• (Womit) Die Studenten lernen das Ineinandergreifen verschiedener wissenschaftlicher Arbeitsbereiche kennen wie Logik, Algorithmik, Kognitionswissenschaften, Informatik.</li> <li>• (Womit) Sie erhalten einen Überblick über das von unterschiedlichen Disziplinen beeinflusste und in vielen Softwaresystemen angewendete Wissensgebiet Künstliche Intelligenz.</li> <li>• (Womit) Sie lernen Inhalte, Methoden, Lösungsansätze, Sprachen und Werkzeuge der KI und werden mit den Arbeitsgebieten der KI vertraut.</li> <li>• (Wozu) Das vermittelte Grundwissen soll den Studenten ermöglichen, Problemstellungen aus den Arbeitsgebieten der KI, z.B. Intelligenten Agenten, in allen wissenschaftlichen Dimensionen zu erfassen und an Lösungen in Projektteams mitzuarbeiten.</li> </ul>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Agenten und Agentenumgebungen</li> <li>• Problemlösen</li> <li>• uninformierte, informierte und adversariale Suchstrategien</li> <li>• Constraint Satisfaction Probleme</li> <li>• Adversiale Suche</li> <li>• Min-Max-Algorithmus</li> <li>• Alpha-/Beta-Kürzung</li> <li>• Grundlagen Genetische Algorithmen</li> <li>• Grundlagen Neuronale Netze</li> <li>• Bayes-Theorem - Entscheiden unter Unsicherheit</li> <li>• Bayes-Belief Networks, Sampling</li> <li>• Bayes-Filter. Wahrscheinlichkeitsverteilungen unbeobachteter Zustände.</li> <li>• Grundlagen der Monte Carlo Lokalisierung</li> <li>• Lokalisierung durch Partikelfilter</li> <li>• Hidden Markov Models. Dynamische Bayes-Netze.</li> </ul>
Lehr- und Lernmethoden:	Folien und handschriftliche Unterlagen. 4 SWS: 2 SWS Vorlesung + 2 SWS Praktikum
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	Mathematik, Algorithmen und Programmierung I+II und Algorithmik erfolgreich abgeschlossen.
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Stuart Russell, Peter Norvig. Artificial Intelligence: A Modern Approach, Global Edition 3. Ed. Addison Wesley 2016</li> <li>• Joachim Hertzberg. Mobile Roboter: Eine Einführung aus Sicht der Informatik. Springer 2012</li> <li>• Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic Robotics - Intelligent Robotics and Autonomous Agents. MIT Press 2005</li> <li>• David L. Poole, Alan K. Mackworth. Artificial Intelligence: Foundations of Computational Agents. Oxford University Press. 2010</li> <li>• Stephen Marsland. Machine Learning: An Algorithmic Perspective. Chapman and Hall/CRC, 1. Ed., 2009</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	-
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

**6.19 [Mensch-Computer-Interaktion]**

Modulnummer:	MCI
Modulbezeichnung:	Mensch-Computer-Interaktion
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	4
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Heinrich Klocke
Dozierende:	Prof. Dr. Heinrich Klocke
Learning Outcome:	<p>Cognitive Engineering</p> <ul style="list-style-type: none"> <li>• (Was) Kennenlernen und Begreifen grundlegender kognitiver Fähigkeiten und Grenzen des Menschen.</li> <li>• (Was) Wissen über Gesetzmäßigkeiten und Besonderheiten menschlicher Sinneswahrnehmung und kognitiver Wahrnehmung.</li> <li>• (Was) Leistung des menschlichen Gedächtnisses einschätzen.</li> <li>• (Womit) Verstehen der Mechanismen, die beim Menschen für induktives und deduktives Schließen als Grundlage für Handlungen und Entscheidungen verantwortlich sind</li> <li>• (Womit) Überblick gewinnen über alle Aspekte der Nutzung von Computer-Systemen im Arbeits- und Privatleben</li> <li>• (Womit) Umsetzen von Wissen über kognitive Fähigkeiten bei der Gestaltung von Mensch-Computer-Schnittstellen insbesondere bei Systemen, die in alltägliche Handlungsprozesse integriert sind (ubiquitous computing).</li> </ul> <p>Usability Engineering</p> <ul style="list-style-type: none"> <li>• (Was) Kennlernen der Methodik des szenariobasierten Usability Engineering: Analyse, Entwurf, Entwicklung und Evaluation gebrauchstauglicher Software im Nutzungskontext.</li> <li>• (Womit) Aktives Auseinandersetzen mit der Frage: „Wie kann das Wissen über die kognitiven Fähigkeiten u. Grenzen des Menschen benutzt werden, um gebrauchstaugliche Menschen-Computer Schnittstellen zu gestalten?“</li> <li>• (Womit) Umsetzung der Methodik in konkreten Praktikumsprojekten in Teamarbeit</li> <li>• Umgang mit Zielkonflikten bei der Gestaltung von User Interfaces</li> <li>• (Was) Fähigkeit, etwas auf Benutzersicht und nicht aus Entwicklersicht zu beurteilen</li> <li>• (Womit) Präsentation von Projektergebnissen für die Benutzerzielgruppe</li> <li>• (Womit) Die Studierenden sollen insbesondere motiviert und ermuntert werden, kreative Lösungen bei der Gestaltung von Mensch-Computer-Schnittstellen zu entwickeln. Es soll ein Bewusstsein dafür aufgebaut werden, dass durch neue kreative Ideen mehr Fortschritt erreicht und Neues entdeckt werden kann, als durch das Gehen bekannter und gewohnter Wege.</li> </ul> <p>Kompetenz: (Wozu) Innovation durch Kreativität. Offenheit für neue Ideen bei der Gestaltung von Mensch-Computer Schnittstellen. Tiefes Verständnis für den engen Zusammenhang zwischen kognitiven Prozessen des Benutzers und der Gestaltung (multimodaler) Information und Interaktion.</p>
Modulinhalte:	<p>Cognitive Engineering</p> <ul style="list-style-type: none"> <li>• Sinneswahrnehmung (sensation), kognitive Wahrnehmung (perception), Wahrnehmungskanäle</li> <li>• Modelle der Informationsverarbeitung</li> <li>• Gedächtnis: <ul style="list-style-type: none"> <li>• Arten: implizit, explizit, episodisch</li> <li>• Prozesse (Kurz-/Langzeit)</li> <li>• Grenzen und Kapazität (7±2-Regel)</li> </ul> </li> <li>• Modelle: Waugh&amp;Norman, Atkison&amp;Shiffrin, LOR, LOP, Rumelhart&amp;McClelland (PDP), ...</li> <li>• Wahrnehmung und (Welt-)Wissen</li> <li>• Selektive und visuelle Aufmerksamkeit</li> <li>• Bewusstsein: automatische und kontrollierte kognitive Prozesse</li> <li>• Denkprozesse: Problemlösen, Kreativität, menschliche Intelligenz</li> </ul>

	Usability Engineering (Schwerpunkt des Praktikums) <ul style="list-style-type: none"> <li>• Gebrauchstauglichkeit im Nutzungskontext</li> <li>• Interaktionsstile</li> <li>• Requirementsanalysis: Kontext- und Problemszenarien</li> <li>• Activity- und Informationdesign, GOMS-Modelle</li> <li>• Interaction-Design</li> <li>• Usability Evaluation und Eyetracking</li> </ul>
Lehr- und Lernmethoden:	<ul style="list-style-type: none"> <li>• Beamergetützte Vorlesungen (Folien im Netz)</li> <li>• Vertiefende Unterlagen und aktuelle Artikel aus Fachzeitschriften (i-com, interactions)</li> <li>• Praktikapräsenz in Kleingruppen (2-3 Teams á 2-4 Studierende), um aktuelle Arbeitsergebnisse vorzustellen und zu diskutieren. Seminar-raum mit Beamer, WLAN-Netz, Whiteboards/Tafeln</li> <li>• Usability-Tests im Usability-Studio (<a href="http://www.software-quality.fh-koeln.de/usability/">http://www.software-quality.fh-koeln.de/usability/</a>)</li> </ul>
	4 SWS: 2 SWS Vorlesung + 2 SWS Praktikum
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	keine über die Zulassungsbedingungen hinausgehenden Voraussetzungen
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Anderson, J.R. Cognitive Psychology and Its Implications 7. Ed. . Palgrave Macmillan 2009</li> <li>• Coren, S., Ward, L.M., Enns, J.E. Sensation and Perception. 6th ed., Wiley 2004</li> <li>• Dix, A., Finlay, J., Abowd, G., Beale, R. Human-Computer Interaction. 3rd ed. Prentice Hall, 2004</li> <li>• Goldstein, E.B. Cognitive Psychology. Connecting Mind, Research, and Everyday Experience. Thomson Wadsworth, 2005</li> <li>• Krug, Steve. Don't make me think!: Web Usability: Das intuitive Web, mitp 2006</li> <li>• Shneiderman, B., Plaisant, C. Designing the User Interface. 5th ed. Addison-Wesley 2009</li> <li>• Rosson, M.B., Carroll, J.M. Usability Engineering. Morgan Kaufmann Publ. 2002</li> <li>• Solso, R.L., Maclin, M.K., Maclin, O.H. Cognitive Psychology. 7th ed. Pearson, 2005</li> <li>• Ware, C. Information Visualization. Perception for Design. Morgan Kaufmann, 2004</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Wirtschaftsinformatik
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.20 [Paradigmen der Programmierung]

Modulnummer:	PP
Modulbezeichnung:	Paradigmen der Programmierung
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	4



Häufigkeit des Angebots:	Jedes Semester
Modulverantwortliche*r:	Prof. Dr. Christian Kohls
Dozierende:	Alexander Dobrynin, MSc.; Prof. Dr. Christian Kohls
Learning Outcome:	(Was) Die Studierenden sollen unterschiedliche Programmierparadigmen verstehen und anwenden können. Weiterhin sollen sie die Angemessenheit der verschiedenen Programmierparadigmen für eine Aufgabenstellung einordnen und bewerten können. (Womit) Dies geschieht, indem verschiedene Programmiersprachen eingeführt und praktisch ausprobiert werden. Dabei wird vergleichend auf die verschiedenen Ansätze (prozedural-funktional-objektorientiert; deklarativ-imperativ; textuell-visuell; Nebenläufigkeit; idiomatische Strukturen und Muster) eingegangen, indem Algorithmen in den verschiedenen Programmiersprachen umgesetzt werden. (Wozu) Studierende sollen mithilfe von etablierten Paradigmen und Entwurfsmustern in der Lage sein, synchrone und asynchrone Programme zu konzipieren und ablaufsicher zu gestalten.
Modulinhalte:	<ul style="list-style-type: none"> <li>• Grundlagen von Programmiersprachen</li> <li>• Vergleich imperativer und deklarativer Paradigmen</li> <li>• prozedurale und objektorientierte Programmierung (z.B. Kotlin, Java, IO, JavaScript)</li> <li>• funktionale Programmierung (z.B. Kotlin, Scala)</li> <li>• Logikprogrammierung (z.B. Prolog)</li> <li>• Nebenläufigkeit (Thread-Sicherheit, Synchronisation, Deadlocks, Threads, Koroutinen)</li> <li>• Entwurfsmuster (u.a. Kompositum, Beobachter, Strategie, Dekorierer, Iterator)</li> <li>• Visuelle Programmierung</li> </ul>
Lehr- und Lernmethoden:	<ul style="list-style-type: none"> <li>• Vorlesung mit interaktiven Phasen, Präsentationen und Live-Coding</li> <li>• Übung</li> <li>• Praktikum</li> <li>• Selbststudium mit bereitgestellten Screencasts, einem umfassenden Skript sowie Fachliteratur</li> </ul> <p>4 SWS: 2 SWS Vorlesung + 1 SWS Übung + 1 SWS Praktikum</p>
Prüfungsformen:	60 minütige Klausur sowie erfolgreiche Teilnahme am Praktikum
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150 h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	Kenntnis der prozeduralen und der Objektorientierten Programmierung (AP1 und AP2)
Zwingende Voraussetzungen:	keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Skript (<a href="http://www.gm.fh-koeln.de/ehses/paradigmen/">www.gm.fh-koeln.de/ehses/paradigmen/</a>)</li> <li>• W.F. Clocksin, C.S. Mellish (2003). Programming in Prolog. Springer-Verlag</li> <li>• Tate, B. A., &amp; Klicman, P. (2011). Sieben Wochen, sieben Sprachen: Verstehen Sie die modernen Sprachkonzepte. O'Reilly.</li> <li>• Gamma, E., Helm, R., Johnson, R., &amp; Vlissides, J. (2015). Design patterns: Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software. Mitp.</li> <li>• Dmitry Jemerov &amp; Svetlana Isakova (2017). Kotlin in Action. Manning Publications.</li> <li>• Pierre-Yves Saumont (2019). The Joy of Kotlin. Manning Publications.</li> <li>• Dawn Griffiths, David Griffiths &amp; Jørgen W. Lang (2019). Kotlin von Kopf bis Fuß: Eine Einführung in die Kotlin-Programmierung. O'Reilly.</li> <li>• Thomas Theis (2019). Einstieg in Kotlin: Apps entwickeln mit Android Studio. Keine Vorkenntnisse erforderlich, ideal für Kotlin- Einsteiger und Java-Umsteiger. Rheinwerk-Verlag</li> <li>• Karl Szwillus (2019). Kotlin: Einstieg und Praxis. mitp Professional.</li> <li>• Kohls, C., Dobrynin, A., Leonard, F. (2020). Programmieren lernen mit Kotlin. München: Hanser Verlag.</li> <li>• Online-Referenz und Tutorials: <a href="https://kotlinlang.org/docs/reference/">https://kotlinlang.org/docs/reference/</a></li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	Medieninformatik
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.21 [Softwaretechnik II]

Modulnummer:	ST2
Modulbezeichnung:	Softwaretechnik II
Art des Moduls:	Pflicht
ECTS credits:	5
Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	4
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Stefan Bente
Dozierende:	Prof. Dr. Stefan Bente
Learning Outcome:	<p>(Was) Als SW-Entwickler oder SW-Architekt kann ich eine Softwarearchitektur für eine gegebene Aufgabenstellung iterativ erarbeiten und in jeweils passender UMLNotation dokumentieren,</p> <p>(Womit) indem ich</p> <ul style="list-style-type: none"> <li>• <b>(1)</b> sinnvolle <b>Architekturstil(e)</b> für mein IT-System auswähle,</li> <li>• <b>(2.1)</b> die <b>Datenzugriffsschicht</b> nach Grundsätzen des Domain Driven Design, inklusive OR-Mapping, gestalte,</li> <li>• <b>(2.2)</b> die <b>Fachlogik</b> in Form von Funktions-, Prozess- und Bestandskomponenten modelliere,</li> <li>• <b>(2.3)</b> zum Datenaustausch mit Clients sowie anderen Komponenten und Umsystemen eine <b>Serviceschicht</b> mit jeweils passenden Kommunikationsprotokollen spezifiziere,</li> <li>• <b>(2.4)</b> die <b>Präsentationsschicht</b> gemäß aktueller Web-, Mobil- und sonstiger Clienttechnologie modelliere, mein IT-System in eine als Service-orientierte Architektur (SOA) oder Microservices strukturierte <b>IT-Landschaft integriere</b>,</li> </ul> <p>(Wozu) so dass eine passende, langlebige, änderungsfähige Architektur entsteht, die in eine nachhaltig wartbare Software umgesetzt werden kann.</p>
Modulinhalte:	<ul style="list-style-type: none"> <li>• Einführung und Überblick</li> <li>• Begriff der Architekturstile</li> <li>• Schichtenarchitektur</li> <li>• Datenzugriffsschicht</li> <li>• Logikschicht und Komponententypen</li> <li>• Serviceschicht</li> <li>• Interface-Kategorien und REST</li> <li>• Andere Kommunikationsprotokolle (synchron / asynchron)</li> <li>• Präsentationsschicht</li> <li>• Integration von Softwaresystemen</li> <li>• SOA</li> <li>• Microservices</li> </ul>
Lehr- und Lernmethoden:	Vorlesung, Praktikum mit Programmierung 4 SWS: 2 SWS Vorlesung + 2 SWS Praktikum
Prüfungsformen:	Klausur
Workload (25 - 30 h $\hat{=}$ 1 ECTS credit) :	150h
Präsenzzeit:	72h
Selbststudium:	78h
Empfohlene Voraussetzungen:	Algorithmen und Programmierung I und II , Theoretische Informatik
Zwingende Voraussetzungen:	Keine

Empfohlene Literatur:

- Balzert, Helmut. *Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb*. 3. Aufl. 2012. Heidelberg: Spektrum Akademischer Verlag, 2011.
- Booch, Grady, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Cullen, and Kelli A. Houston. *Object-Oriented Analysis and Design with Applications*. 3rd ed. The Addison-Wesley Object Technology Series. Upper SaddleRiver, NJ: Addison-Wesley, 2007.
- Capgemini. (2015). *Architecture Guidelines for Application Design v2.0*.
- Dustdar, Schahram, Harald Gall, and Manfred Hauswirth. *Software-Architekturen für Verteilte Systeme: Prinzipien, Bausteine und Standardarchitekturen für moderne Software*. 1st ed. Springer Berlin Heidelberg, 2003.
- Eilebrecht, Karl, and Gernot Starke. *Patterns kompakt: Entwurfsmuster für effektive Software-Entwicklung*. 4th ed. Berlin: Springer Vieweg, 2013.
- Gregor Engels, Andreas Hess, Bernhard Humm, Oliver Juwig, Marc Lohmann, Jan-Peter Richter, Markus Voß, and Johannes Willkomm. *Quasar Enterprise: Anwendungslandschaften serviceorientiert gestalten*. 1., Aufl. dpunkt.verlag, 2015.
- Evans, Eric. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. 1 edition. Boston: Addison-Wesley Professional, 2003.
- Gamma, Erich, Richard Helm, Ralph E. Johnson, and John Vlissides. *Design Patterns. Elements of Reusable Object-Oriented Software*. 1st ed., Reprint. Reading, Mass: Prentice Hall, 1994.
- Goll, Joachim. *Architektur- und Entwurfsmuster der Softwaretechnik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014. <http://link.springer.com/10.1007/978-3-658-05532-5>.
- Hohpe, Gregor, and Bobby Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. 01 ed. Boston: Addison Wesley, 2003. Auch als frei verfügbarer, sehr empfehlenswerter Blog, der die wesentlichen Informationen beinhaltet: <http://www.enterpriseintegrationpatterns.com/index.html>.
- Jackson, Michael. *Problem Frames & Methods: Analysing and Structuring Software Development Problems*. Harlow: Addison-Wesley Longman, Amsterdam, 2000.
- Lilienthal, Carola. *Langlebige Software-Architekturen: Technische Schulden analysieren, begrenzen und abbauen*. 1st ed. dpunkt.verlag GmbH, 2015.
- Massé, M. (2011). *REST API Design Rulebook* (1. Aufl.). Beijing: O'Reilly and Associates.
- OASP. "Open Application Standard Platform for Java V1.5.0," 2015. <http://oasp.github.io/oasp4j/1.5.0/OASP4J.pdf>.
- Reussner, Ralf, Wilhelm Hasselbring. *Handbuch der Software-Architektur*. 2. Aufl. Heidelberg: dpunkt Verlag, 2008.
- Siedersleben, Johannes. *Moderne Software-Architektur: Umsichtig planen, robust bauen mit Quasar*. 1st ed. Heidelberg: dpunkt, 2004.
- Toth, Stefan. *Vorgehensmuster für Softwarearchitektur: Kombinierbare Praktiken in Zeiten von Agile und Lean*. 2., aktualisierte und erweiterte Auflage. München: Carl Hanser Verlag GmbH & Co. KG, 2015.
- Tilkov, Stefan, Martin Eigenbrodt, Silvia Schreier, and Oliver Wolf. *REST und HTTP: Entwicklung und Integration nach dem Architekturstil des Web*, p. 11ff. 3. Aufl. Heidelberg: dpunkt.verlag GmbH, 2015.
- OMG. "OMG Unified Modeling Language, Version 2.5," 2015. <http://www.omg.org/spec/UML/2.5/>.
- Winter, Mario. *Vorlesung Softwaretechnik in der Medieninformatik*. SS 2015, TH Köln.
- Wolff, Eberhard. *Microservices: Grundlagen flexibler Softwarearchitekturen*. 1., Auflage. dpunkt.verlag, 2015.
- Zörner, Stefan. *Softwarearchitekturen dokumentieren und kommunizieren: Entwürfe, Entscheidungen und Lösungen nachvollziehbar und wirkungsvoll festhalten*. 2., Aufl. München: Carl Hanser Verlag, 2015.

Verwendung des Moduls in weiteren Studiengängen:	-
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

**6.22 [Spezielle Gebiete der Datenbanken]**

Modulnummer:	SGDB
Modulbezeichnung:	Spezielle Gebiete der Datenbanken
Art des Moduls:	Pflicht
ECTS credits:	5

Sprache:	Deutsch
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	4
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Prof. Dr. Birgit Bertelsmeier
Dozierende:	Prof. Dr. Birgit Bertelsmeier, NN (Nachfolge Prof. Dr. Heide Faeskorn-Woyke)
Learning Outcome:	<p>Die Studierenden sollen:</p> <ul style="list-style-type: none"> <li>• (Was) Konzepte der Datenbankanwendungsprogrammierung und aktiver Datenbanken verstanden haben und in der Lage sein, für Aufgabenstellungen Lösungen zu entwickeln, zu programmieren und zu bewerten.</li> <li>• (Womit) die Basisfunktionalitäten von Aufrufchnittstellen kennen und sie beispielhaft für aktuelle Programmiersprachen anwenden können.</li> <li>• (Wozu) um die Besonderheiten, Schwierigkeiten der Verwendung von Datenbanksystemen im World Wide Web wissen und Strategien zu deren Handling kennen und verstehen und dies im relationalen wie auch NoSQL-Kontext.</li> <li>• (Was) über ein Verständnis der drei Datenbankarten: relational, objektrelational und objektorientiert verfügen, insbesondere ihre Gemeinsamkeiten, Unterschiede kennen sowie Abbildungsstrategien zwischen ihnen anwenden können.</li> <li>• (Womit) grundlegende Speicherstrukturen kennen hinsichtlich ihrer Anwendungsspezifika bewerten können.</li> <li>• (Womit) ein Verständnis für die Notwendigkeit von nicht relationalen DBS-Konzepten entwickeln sowie alternative NoSQL-Konzepte und -Kategorien mit ihren Vor- und Nachteilen sowie bevorzugten Einsatzgebieten kennen.</li> <li>• (Wozu) insbesondere vor dem Hintergrund der enorm anwachsenden Datenmengen und neuen Verknüpfungsformen in der Lage sein, Daten kritisch zu hinterfragen und verantwortungsvoll mit ihnen umzugehen.</li> </ul>
Modulinhalte:	<p>Einführung in die Datenbankanwendungsprogrammierung mit einem Überblick über die alternativen Konzepte und -architekturen und deren Bewertung für verschiedene Anwendungskontexte.</p> <p>Aktive Datenbanksysteme mit ihren Funktionen werden anhand des aktuellen SQL-Standard (SQL2016) vermittelt und deren praktische Umsetzung wird mit Hilfe von Oracle PL-SQL, Oracle-Trigger bzw. MySQL-stored routines und -Trigger erarbeitet.</p> <p>Für die DBS Oracle bzw. MySQL werden Verbindungen zu verschiedenen aktuellen Programmiersprachen aufgebaut (z.B. Java, php, Kotlin, ...)</p> <p>Die Anforderungen an den Einsatz von DBS im Web: wie z.B. Interaktivität, Verfügbarkeit, Performance, Skalierung, Sicherheit erfordern spezielle Konzepte für u.a. Handling von Sessions, Transaktionen, Datenkorrektheit und -sicherheit. Aktuelle Techniken zur Anbindung von DBS im WWW werden vorgestellt, wie auch die Basisstrategien für statische und dynamische (client-/serverseitige) Web-Seiten auf dem Client.</p> <p>Grundbegriffe und Architekturen relationaler, objektrelationaler und objektorientierter Datenbanken im Vergleich und Strategien zur Transformation zwischen relationalen und objektorientierten Konzepten. Einen Einblick in die Datenbanksprache SQL mit ihren objektrelationalen Erweiterungen bei Oracle</p> <p>Ein Überblick über gängige Speicherstrukturen mit ihren Einsatzgebieten sowie deren Realisierung bei Oracle.</p> <p>Eine Motivation der NoSQL-Idee, ihrer zentralen Konzepte und die aktuellen NoSQL-Kategorien wie Key/Value, Wide Cols, Documents, Graph, In Memory im Überblick und im Vergleich. Zudem alternative Verfahren wie u.a. MVCC, BASE, Skalierungen.</p>
Lehr- und Lernmethoden:	<p>Inverted Classroom, Folien und aufgezeichnete Vorlesungen und Übungen</p> <p>eLearning-Datenbank-Portal der TH Köln mit interaktivem PL/SQL- und Trigger-Trainern und Multiple-Choice-Tests (<a href="https://edb2.gm.thkoeln.de/">https://edb2.gm.thkoeln.de/</a>)</p> <p>Lernvideos von Prof. Dr., Jens Dittrich, Universität des Saarlandes, Fakultät für Mathematik und Informatik: <a href="https://www.youtube.com/user/jensdit">https://www.youtube.com/user/jensdit</a> und <a href="https://bigdata.uni-saarland.de/datenbankenlernen">https://bigdata.uni-saarland.de/datenbankenlernen</a>)</p> <p>4 SWS: 2 SWS Vorlesung + 1 SWS Übung + 1 SWS Praktikum</p>

Prüfungsformen:	Erfolgreiche Teilnahme am Praktikumsprojekt als Voraussetzung für die Zulassung zur Klausur Klausur
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	150h
Präsenzzeit:	45h
Selbststudium:	105h
Empfohlene Voraussetzungen:	Das Wissen und die Fähigkeiten, wie sie in den Modulen Datenbanksysteme sowie in Algorithmen und Programmierung I und II vermittelt werden.
Zwingende Voraussetzungen:	Keine
Empfohlene Literatur:	<ul style="list-style-type: none"> <li>• Damaschke, G.: "PHP und MySQL - Der Web-Baukasten für Einsteiger und Individualisten", 2015, 1. Auflage, Wiley-VCH, 978-3-527-69264-4, (eBook)</li> <li>• Edlich, St., Friedland, A., Hampe, J., Brauer, B., Brückner, M.: "NoSQL - Einstieg in die Welt nichtrelationaler Web 2.0</li> <li>• Datenbanken", 2011, 2. Auflage, Hanser, 978-3-446-42753-2, (eBook)</li> <li>• Grimm, P., Keber, T. O., Zöllner, O. (Hrsg.): "Digitale Ethik - Leben in vernetzten Welten" 2019, Reclam, Philipp, 978-3-15-015240-9, (eBook)</li> <li>• Gull, C.: "Web-Applikationen entwickeln mit NoSQL", 2012, Franzis Verlag, 978-3-645-22022-4, (eBook)</li> <li>• Heuer, A., Saake, G., Sattler, K.-U., Grunert, H., Meyer, H.: "Datenbanken - Praxiswissen für Anwender", 2020, MITP, 978-3-95845-782-9, (eBook)</li> <li>• Kaur, M., Shaik, B.: "PostgreSQL Development Essentials", 2016, Packt Publishing, 978-1-78398-901-0, (eBook)</li> <li>• Kudraß, Th. (Hrsg.): "Taschenbuch Datenbanken", 2015, 2. Auflage, Carl Hanser, 978-3-446-43508-7, (eBook)</li> <li>• Saake, G., Sattler, K.-U., Heuer, A.: "Datenbanken - Implementierungstechniken", 2019, 4. Auflage, MITP, 978-3-95845-779-9, (eBook)</li> <li>• Sieben, J.: "Oracle APEX - Das umfassende Handbuch für Entwickler mit PL/SQL-Kenntnissen", 2017, 1. Auflage, Rheinwerk, 978-3-836-24530-2, (eBook)</li> <li>• Sieben, J.: "Oracle PL/SQL - Das umfassende Handbuch für Datenbankentwickler", 2017, 3. Auflage, Rheinwerk, 978-3-836-26073-2, (eBook)</li> <li>• Solmecke, Ch., Kocatepe, S.: "DSGVO für Website-Betreiber - Ihr Leitfaden für die sichere Umsetzung der EU-Datenschutz-Grundverordnung", 2018, 2. Auflage, Rheinwerk Computing, 978-3-8362-6714-4, (eBook)</li> <li>• Theis, Th.: "Einstieg in PHP 7 und MySQL - Für Programmieranfänger geeignet. So programmieren Sie dynamische Websites mit PHP und MySQL. Inkl. MariaDB", 2018, 13. Auflage, Rheinwerk, 978-3-836-26312-2, (eBook)</li> </ul>
Verwendung des Moduls in weiteren Studiengängen:	
Besonderheiten:	Keine
Letzte Aktualisierung:	20.10.2020

## 6.23 [Praxissemester]

Modulnummer:	PS
Modulbezeichnung:	Praxissemester
Art des Moduls:	Pflichtmodul
ECTS credits:	30
Sprache:	Abhängig vom Unternehmen
Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	5
Häufigkeit des Angebots:	Jedes Semester

Modulverantwortliche*r:	Praxissemesterbeauftragte/r (N.N.)
Dozierende:	Dozentinnen und Dozenten der Lehreinheit Informatik
Learning Outcome:	Das Praxissemester soll die Studierenden an die berufliche Tätigkeit der „Informatikerin“ oder des „Informatikers“ durch konkrete Aufgabenstellung und praktische Mitarbeit in Betrieben oder anderen Einrichtungen der Berufspraxis heranzuführen. Die Studierenden sollen in der Lage sein, die im bisherigen Studium erworbenen Kenntnisse und Fähigkeiten anzuwenden und die bei der praktischen Tätigkeit gemachten Erfahrungen zu reflektieren und auszuwerten.
Modulinhalte:	Im Praxissemester werden die Studierenden durch eine ihrem Ausbildungsstand angemessene Aufgabe mit den wissenschafts- und berufsfeldbezogenen Arbeitsweisen in der Informatik vertraut gemacht. Sie sollen diese Aufgabe nach entsprechender Einführung selbstständig, allein oder in der Gruppe unter fachlicher Anleitung bearbeiten.
Lehr- und Lernmethoden:	Kenntnisse aus dem bisherigen Studium werden in der Praxis angewendet. Schlüsselqualifikationen zu effektiver und teamorientierter Arbeit im betrieblichen Umfeld werden umgesetzt. Eigene Arbeiten und Ergebnisse werden beurteilt, präsentiert und einem Auditorium erläutert.
Prüfungsformen:	Nach Abschluss des Praxissemesters wird dem betreuenden Mentor ein wissenschaftlicher Praxissemesterbericht sowie das Zeugnis der Praxisstelle zur Anerkennung vorgelegt. Der Bericht wird nicht benotet.
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	900 h
Präsenzzeit:	22 Wochen in Vollzeit im Unternehmen/Forschungsprojekt
Selbststudium:	...
Empfohlene Voraussetzungen:	
Zwingende Voraussetzungen:	Auf Antrag wird zum Praxissemester zugelassen, wer alle Fächer der Studienabschnitte Grundlagen bestanden hat. Siehe Praxissemesterordnung (Anhang der Prüfungsordnung).
Empfohlene Literatur:	
Verwendung des Moduls in weiteren Studiengängen:	Wirtschaftsinformatik, IT-Management
Besonderheiten:	<p>Das Praxissemester wird in der Regel in Industrieunternehmen, Forschungseinrichtungen oder entsprechend ausgestatteten Behörden durchgeführt.</p> <p>Das Praxissemester kann in Ausnahmefällen auch an der Hochschule selbst z.B. im Rahmen eines Forschungsprojekts absolviert werden.</p> <p>Ein Praxissemester im Ausland ist besonders erwünscht.</p> <p>Ein Studiensemester im Ausland ist möglich.</p> <p>Die Studierenden bewerben sich selbstständig um eine Praxisstelle.</p> <p>Die Studierenden können an Prüfungen während des Praxissemesters teilnehmen.</p>
Letzte Aktualisierung:	20.10.2020

## 6.24 [Wahlspezialisierung I]

Modulnummer:	WASP I
Modulbezeichnung:	Wahlspezialisierung I
Art des Moduls:	Wahlpflicht
ECTS credits:	15
Sprache:	Deutsch

Dauer des Moduls:	1 Semester
Empfohlenes Studiensemester:	6
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Studiengangsleiter*in des Studiengangs Informatik
Dozierende:	Lehrende der Lehreinheit Informatik
Learning Outcome:	Ziel ist die Förderung von Problemlösungskompetenz, Selbstmanagementfähigkeit sowie der Fähigkeit zum selbstständigen wissenschaftlichen Arbeiten. Das Projekt dient außerdem der Wissensvertiefung über Fächer- und Schwerpunktgrenzen hinweg durch praktische Erfahrungen.
Modulinhalte:	<p><b>Fachliche Grundlagenveranstaltung (5CP):</b> Der Dozent liefert die für den Projektteil nötigen Grundlagen und Kenntnisse. Hierzu gehören unter Anderem fachliche Grundlagen, bestehende Forschungsergebnisse, Domänenwissen, etc.</p> <p><b>Projektteil (10CP):</b> Bearbeitung einer komplexeren Aufgabenstellung im Team, im Rahmen eines Forschungs- und Entwicklungsprojekts (ggf. auch in Kooperation mit externen Partnern). Der Dozent definiert die Zielsetzung und führt einen regelmäßigen Diskurs über den Fortgang des Projekts. Kommunikations- und Kooperationsformen werden vorab gemeinsam zwischen Dozent*innen und Studierenden festgelegt und in regelmäßigen Abständen gemeinsam reflektiert.</p> <p><b>Fachliche Grundlagenveranstaltung und Projektteil</b> können in Kooperation von 2 und mehr Dozierenden angeboten werden.</p>
Lehr- und Lernmethoden:	Legt die Dozent*in vor der Veranstaltung fest.
Prüfungsformen:	Legt die Dozent*in vor der Veranstaltung fest.
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	450h
Präsenzzeit:	200h
Selbststudium:	250h
Empfohlene Voraussetzungen:	Alle Module aus den Grundlagen (1. und 2. Semester) und der Vertiefung (3. und 4. Semester) müssen abgeschlossen sein.
Zwingende Voraussetzungen:	Fachlich: Je nach Aufgabenstellung
Empfohlene Literatur:	-
Verwendung des Moduls in weiteren Studiengängen:	-
Besonderheiten:	-
Letzte Aktualisierung:	20.10.2020

## 6.25 [Wahlspezialisierung II]

Modulnummer:	WASP II
Modulbezeichnung:	Wahlspezialisierung II
Art des Moduls:	Wahlpflicht
ECTS credits:	15
Sprache:	Deutsch
Dauer des Moduls:	1 Semester

Empfohlenes Studiensemester:	6
Häufigkeit des Angebots:	Jedes Sommersemester
Modulverantwortliche*r:	Studiengangsleiter*in des Studiengangs Informatik
Dozierende:	Alle Lehrenden der Lehrinheit Informatik
Learning Outcome:	Ziel ist die Förderung von Problemlösungskompetenz, Selbstmanagementfähigkeit sowie der Fähigkeit zum selbstständigen wissenschaftlichen Arbeiten. Das Projekt dient außerdem der Wissensvertiefung über Fächer- und Schwerpunktgrenzen hinweg durch praktische Erfahrungen.
Modulinhalte:	<p><b>Alternative I:</b> Das Modul WASP II kann aus bis zu drei Teilmodulen á 5 ECTS, die von den Studierenden aus dem bestehenden Wahlpflichtangeboten gewählt werden können, bestehen.</p> <p><b>Alternative II:</b> Das Modul WASP II kann aus zwei Teilmodulen bestehen: Teilmodul mit 5 ECTS, das von den Studierenden aus dem bestehenden Wahlpflichtangebot gewählt werden kann Teilmodul mit 10 ECTS, der einen höheren praktischen Anteil beinhalten kann.</p> <p><b>Alternative III:</b> Alle Lehrangebote, deren Beschreibung der Modulbeschreibung <a href="#">WASP I</a> entsprechen.</p>
Lehr- und Lernmethoden:	Legt die Dozent*in vor der Veranstaltung fest.
Prüfungsformen:	Legt die Dozent*in vor der Veranstaltung fest.
Workload (25 - 30 h $\triangleq$ 1 ECTS credit) :	450h
Präsenzzeit:	Variabel
Selbststudium:	Variabel
Empfohlene Voraussetzungen:	
Zwingende Voraussetzungen:	Alle Module aus den Grundlagen (1. und 2. Semester) und der Vertiefung (3. und 4. Semester) müssen abgeschlossen sein.
Empfohlene Literatur:	Fachlich: Je nach Aufgabenstellung
Verwendung des Moduls in weiteren Studiengängen:	-
Besonderheiten:	-
Letzte Aktualisierung:	20.10.2020

## 6.26 [Praxisprojekt mit begleitendem Projektseminar]

Modulnummer:	PrPr
Modulbezeichnung:	Praxisprojekt mit begleitendem Projektseminar
Art des Moduls:	Pflicht
ECTS credits:	15
Sprache:	Deutsch oder Englisch
Dauer des Moduls:	½ Semester
Empfohlenes Studiensemester:	7
Häufigkeit des Angebots:	Jedes Semester



Modulverantwortliche*r:	Dozentinnen und Dozenten der Lehrinheit Informatik
Dozierende:	Dozentinnen und Dozenten der Lehrinheit Informatik
Learning Outcome:	Die Studierenden sollen in der Lage sein Methoden und Techniken, die sie im Studium erlernt haben, in einem realitätsnahen Projekt unter wissenschaftlichen Aspekten weitgehend selbstständig anzuwenden.
Modulinhalte:	Anwendung von Modulinhalten des ersten bis sechsten Semesters anhand von realen Anforderungen in einem praxisrelevanten Kontext.  Dies kann entweder in einem Unternehmen oder in der Hochschule – dann eingebettet in Forschungsprojekte – erfolgen.
Lehr- und Lernmethoden:	Vom betreuenden Dozenten angeleitete selbstständige Arbeit
Prüfungsformen:	Teilnahme an mind. zwei Seminartagen ist Voraussetzung für den eigenen Vortrag an einem 3. Seminartag. Projektdokumentation und Vortrag beim Praxisprojektseminar.  Die Gesamtnote des Praxisprojekts besteht aus zwei Teilnoten (80% und 20%):  Teilnote 1: Projektteil: Dokumentation der Arbeitsergebnisse (80% der Note) Teilnote 2: Seminarteil: Teilnahme an zwei Seminartagen und eigener Seminarvortrag an einem dritten Seminartag (20% der Note)
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	450h
Präsenzzeit:	10h
Selbststudium:	440h
Empfohlene Voraussetzungen:	
Zwingende Voraussetzungen:	Alle Pflichtmodule der Studienabschnitte Grundlagen, Vertiefung und Spezialisierung
Empfohlene Literatur:	je nach Projektthema
Verwendung des Moduls in weiteren Studiengängen:	Wirtschaftsinformatik (BA) Medieninformatik (BA) IT-Management (BA)
Besonderheiten:	
Letzte Aktualisierung:	20.10.2020

## 6.27 [Bachelorarbeit]

Modulnummer:	BA-Thesis
Modulbezeichnung:	Bachelorarbeit
Art des Moduls:	Pflicht
ECTS credits:	12
Sprache:	Deutsch/Englisch
Dauer des Moduls:	9-12 Wochen
Empfohlenes Studiensemester:	7
Häufigkeit des Angebots:	Jedes Semester
Modulverantwortliche*r:	Dozentinnen und Dozenten der Lehrinheit Informatik

Dozierende:	Dozentinnen und Dozenten der Lehreinheit Informatik
Learning Outcome:	Die Bachelorarbeit soll zeigen dass der Prüfling befähigt ist, innerhalb einer vorgegebenen Frist eine praxisorientierte Aufgabe aus seinem Fachgebiet sowohl in ihren fachlichen Einzelheiten als auch in den fachübergreifenden Zusammenhängen nach wissenschaftlichen, fachpraktischen und gestalterischen Methoden selbstständig zu bearbeiten.  Die Bachelorarbeit ist in der Regel eine eigenständige Untersuchung mit einer Aufgabenstellung aus der Informatik und einer ausführlichen Beschreibung und Erläuterung ihrer Lösung.
Modulinhalte:	Das Thema und der Inhalt der Bachelor-Thesis wird in Absprache mit einem betreuenden Professor der Lehreinheit Informatik besprochen und festgelegt.
Lehr- und Lernmethoden:	Die Studierenden bearbeiten die Themenstellung der Bachelor-Thesis selbstständig und werden insbesondere durch die Mitarbeiter des betreuenden Instituts/Labors unterstützt. Ergänzend finden regelmäßige Gespräche mit dem betreuenden Professor statt. Die Bachelor-Thesis kann in der Hochschule oder alternativ bei einem externen Industrieunternehmen durchgeführt werden. In geeigneten Fällen kann sie auch als schriftliche Hausarbeit (Literaturarbeit) erstellt werden.
Prüfungsformen:	Schriftliche Ausarbeitung (Bachelorarbeit)
Workload (25 - 30 h $\cong$ 1 ECTS credit) :	360 h (12 ECTS)
Präsenzzeit:	Regelmäßige Besprechungen mit dem betreuenden Professor
Selbststudium:	
Empfohlene Voraussetzungen:	Erfolgreicher Abschluss aller anderen Module des Studiums
Zwingende Voraussetzungen:	
Empfohlene Literatur:	je nach Thema der Bachelor-Thesis
Verwendung des Moduls in weiteren Studiengängen:	Wirtschaftsinformatik (BA) Medieninformatik (BA) IT-Management (BA)
Besonderheiten:	
Letzte Aktualisierung:	

## 6.28 [Bachelor Kolloquium]

Modulnummer:	BA-Kolloquium
Modulbezeichnung:	Bachelor Kolloquium
Art des Moduls:	Pflicht
ECTS credits:	3
Sprache:	Deutsch / Englisch
Dauer des Moduls:	9 – 12 Wochen (im Anschluss an BA)
Empfohlenes Studiensemester:	7
Häufigkeit des Angebots:	Jedes Semester
Modulverantwortliche*r:	Dozentinnen und Dozenten der Lehreinheit Informatik
Dozierende:	Dozentinnen und Dozenten der Lehreinheit Informatik
Learning Outcome:	Die Studierenden sollen lernen, eine komplexe selbst angefertigte Arbeit in einer vorgegebenen Zeit zu präsentieren.

<b>Modulinhalte:</b>	Der Studierende hält über das Thema der Bachelorarbeit einen Vortrag. Insbesondere sollen Aufgabenstellung, gegebenenfalls der Kontext zu einem Gesamtkonzept des Unternehmens und die eingeschlagenen Lösungswege erläutert werden.  Anschließend findet ein Fachgespräch mit dem 1. und 2. Prüfer statt.
<b>Lehr- und Lernmethoden:</b>	Vortrag und Fachgespräch
<b>Prüfungsformen:</b>	Mündliche Prüfung
<b>Workload (25 - 30 h <math>\hat{=}</math> 1 ECTS credit) :</b>	90h
<b>Präsenzzeit:</b>	30-45 Min für Vortrag und Fachgespräch
<b>Selbststudium:</b>	
<b>Empfohlene Voraussetzungen:</b>	
<b>Zwingende Voraussetzungen:</b>	Erfolgreich abgeschlossene Bachelorarbeit und alle anderen Module des Studiums
<b>Empfohlene Literatur:</b>	je nach Thema der Bachelor-Thesis
<b>Verwendung des Moduls in weiteren Studiengängen:</b>	Wirtschaftsinformatik (BA) Medieninformatik (BA) IT-Management (BA)
<b>Besonderheiten:</b>	
<b>Letzte Aktualisierung:</b>	20.10.2020

## 7 Modulmatrix

Modulmatrix		Studiengang: Informatik Bachelor										Fakultät: Fakultät für Informatik und Ingenieurwissenschaften					Prüfungen	
Module / Lehrveranstaltungen		Zuordnung Kompetenzen Absolvent*innenprofil										Zuordnung Studiengangskriterien					Anzahl	
Semester	Modul	87	51	72	SPR	SQM	INO	PRD	KOM	TPA	WIA	Interdisziplinariät	Digitalisierung	Transfer	Prüfungen	Anzahl		
		IVB	OK	PKA											25			
1	Einführung in die Informatik	3	1	1	x		x		x							0		
	Einführung in Betriebssysteme und Rechnerarchitektur	3	1	1	x		x		x							1		
	Mathematik I	4	2	1	x		x		x			x				1		
	Algorithmen und Programmierung I	4	0	4	x			x						x		1		
2	Theoretische Informatik I	3	0	2	x		x		x							1		
	Mathematik II	5	2	1	x		x		x							1		
	Algorithmen und Programmierung II	3	1	3	x		x		x							1		
	Theoretische Informatik II	3	0	2	x		x		x							1		
	Betriebssysteme und verteilte Systeme	3	1	1	x		x		x							1		
	BWL	0	3	2	x		x		x							1		
	Algorithmik	4	0	1	x				x							1		
3	Projektmanagement	1	2	2	x				x							1		
	Kommunikationstechnik und Netze	2	0	3	x				x							1		
	Softwaretechnik I	3	0	2	x				x							1		
	Datenbanksysteme	2	1	2	x				x							1		
	Informations- und Prozessmanagement	1	0	4	x				x							0		
4	Spezielle Gebiete der Datenbanken	2	1	2	x				x							1		
	Paradigmen der Programmierung	3	1	1	x				x							1		
	Softwaretechnik II	3	0	2	x				x							1		
	Informatik, Recht und Gesellschaft	0	3	2	x				x							1		
	Künstliche Intelligenz	4	0	1	x				x							1		
	Mensch-Computer-Interaktion	1	2	2	x				x							1		
5	Praxissemester	10	10	10	x				x							0		
	Wahlspezialisierung I	5	5	5	x				x							1		
6	Wahlspezialisierung II	5	5	5	x				x							1		
	Praxisprojekt mit begleitendem Projektsseminar	5	5	5	x				x							1		
7	Bachelor	4	4	4	x				x							1		
	Kolloquium	1	1	1	x				x							1		

Impressum:

TH Köln  
Gustav-Heinemann-Ufer 54  
50968 Köln

[www.th-koeln.de](http://www.th-koeln.de)

MDH-Template-Version: 2020-02-10\_V3